AFRL-RI-RS-TR-2008-186
Final Technical Report
June 2008

SPEECHLINKS: ROBUST CROSS-LINGUAL TACTICAL COMMUNICATION AIDS

University of Southern California

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-186 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

SHARON M. WALTER
Work Unit Manager

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUN 2008	Final	Sep 06 – Dec 07

4. TITLE AND SUBTITLE

SPEECHLINKS: ROBUST CROSS-LINGUAL TACTICAL COMMUNICATION AIDS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-06-1-0250

5c. PROGRAM ELEMENT NUMBER
62303E

6. AUTHOR(S)

Shrikanth Narayanan and Panayiotis Georgiou

5d. PROJECT NUMBER
TRAN

5e. TASK NUMBER
TA

5f. WORK UNIT NUMBER
C2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Southern California
University Gardens, Ste 203
Los Angeles CA 90089-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIED
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-186

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-3836

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project was directed toward developing a unique tactical language translator with context-aware, mixed-initiative capability for tactical missions. The specific goals were to: 1) enable robust mixed-initiative tactical communication targeting multiple languages, 2) develop algorithms and tools that enable rapid construction and deployment of systems for new missions and languages, and 3) develop and implement a holistic evaluation process for cross-lingual communication systems that unifies usability, task achievement and cost.

15. SUBJECT TERMS
Language translator, speech translation, speech-to-speech, spoken language translation, context modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Sharon M. Walter
U	U	U	UU	194	19b. TELEPHONE NUMBER (*Include area code*) N/A

Contents

List of Tables

List of Figures

Chapter 1

Overview

In our efforts towards Rapid System Development we have followed a multi-pronged approach. In summary we had developments in

System development In this aspect we worked in both pure software development aspects as well as system implementation techniques that benefit the overall system performance, especially when it comes to new domains and languages. For example:

- System: Built robust speech to speech translation: Signicant progress in constructing, training, and testing our own speech recognition and translation engines. The system was built from the ground up and included our own Speech Recognition, Translation, and Communication engines.
- Robust Translation: Enabled robust translation by systematically employing multiple translation techniques. We employed an ontology based translation system and a statistical machine translation systems. Due to the breadth of the domain, the ontology based translation has proven to be challenging to build in this domain, however recent developments show promising results.
- User Modeling: In our objective of developing an effective user interface for speech mediation, while including user behavior as a parameter in the systems strategy, we have performed several studies that informed our developments. We investigated the benets of a multimodal interface versus a unimodal one; investigated the effects of learning on the user performance through longitudinal studies; implemented and evaluated several forms of system- user interaction; etc.
- Clustered-models: In our experience with the S2S systems in the military environment we have observed a very large variability in the pronunciation patterns of the speakers. The additional data collections help to address these issues, but the fact remains that we need better techniques for addressing the diversity. To do so we have created a speaker-class identification system that can switch acoustic models on the fly.

Research towards better techniques of language modeling, acoustic modeling, user-machine interaction:

- Data Exploitation for rapid domain and language porting: Given todays vast public and digitally available resources we start with the idea that the system can in a semi-automated fashion exploit existing resources. We can thus obtain significant linguistic

knowledge from this approach that improve both the speech recognition and speech translation process.

Performance of statistical n-gram language models depends heavily on the amount of training text material and the degree to which the training text matches the domain of interest. The language modeling community is showing a growing interest in using large collections of text (obtainable, for example, from a diverse set of resources on the Internet) to supplement sparse in-domain resources. However, in most cases the style and content of the text harvested from the web differs significantly from the specific nature of these domains. In **Chapter 2**, we present a relative entropy based method to select subsets of sentences whose n-gram distribution matches the domain of interest. We present results on language model adaptation using two speech recognition tasks: a medium vocabulary medical domain doctor-patient dialog system and a large vocabulary transcription system for European parliament plenary speeches. We show that the proposed subset selection scheme leads to performance improvements over state of the art speech recognition systems in terms of both speech recognition Word Error Rate (WER) and language model perplexity (PPL). Improvements in data selection also translate to a significant reduction in the vocabulary size as well as the number of estimated parameters in the adapted language model.

In addition to learning within language statistical patterns we also want to exploit knowledge to achieve semi-unsupervised part of speech tagging. **Chapter 5** investigates the use of domain knowledge to constrain and improve the unsupervised learning of a classifier, specifically a part-of-speech tagger. We view the contribution of the knowledge source as a reduction in the uncertainty of the model's decisions, quantified by the resulting conditional entropy of the label distribution given the input corpus. We evaluate our approach with increasing levels of knowledge, integrating both hard and soft constraints into a standard Hidden Markov Model (HMM) tagger as virtual evidence (VE). We show improvements of up to 20 or 30 points in percentage accuracy, depending on the method of state-to-label assignment, in addition to more stable and efficient training convergence. We also find that the label entropy induced by the knowledge source is highly predictive of final model performance. Finally, we analyze the problem of mapping the model's internal states to the desired label set, in particular the practical requirements for annotated data in making quality assignments, and the effect of domain knowledge on those requirements.

- Improving the front end of the speech recognizer remains one of the most challenging issues in speech to speech translation. In our work presented in **Chapter 3** we provide the significant benefits derived from employing bio-inspired features for automatic speech recognition based on the early processing stages in the human auditory system. The utility and robustness of the derived features are validated in a speech recognition task under a variety of noise conditions. First, we develop an auditory based feature by replacing the filterbank analysis stage of Mel-frequency cepstral coefficients (MFCC) feature extraction with an auditory model that consists of cochlear filtering, inner hair cell, and lateral inhibitory network stages. Then, we propose a new feature set that retains only the cochlear channel outputs that are more likely to fire the neurons in the central auditory system. This feature set is extracted by principal component analysis (PCA) of nonlinearly compressed early auditory spectrum. When evaluated in a connected digit recognition task using the Aurora 2.0 database, the proposed feature set has *40%* and *18%* average word error rate improvement relative to the MFCC and RelAtive

SpecTrAl (RASTA) features, respectively.

- In User modeling we made gains in both the longitudinal benefits of system usage and multimodal behavior of users. **Chapter 4** addresses modeling user behavior in interactions between two people that do not share a common spoken language and communicate with the aid of an automated bidirectional speech translation system. These interaction settings are complex. The translation machine attempts to bridge the language gap by mediating the verbal communication, noting however that the technology may not be always perfect. In a step toward understanding user behavior in this mediated communication scenario, usability data from doctor-patient dialogs involving a two way English-Persian speech translation system are analyzed. We specifically consider user behavior in light of potential uncertainty in the communication between the interlocutors. We analyze the Retry (*Repeat and Rephrase*) versus Accept behaviors in the mediated verbal channel and as a result identify three user types – *Accommodating*, *Normal* and *Picky*, and propose a dynamic Bayesian network model of user behavior. To validate the model, we performed offline and online experiments. The experimental results using offline data show that correct user type is clearly identified as a user keeps his/her consistent behavior in a given interaction condition. In the online experiment, agent feedback was presented to users according to the user types. We show high user satisfaction and interaction efficiency in the analysis of user interview, video data, questionnaire and log data.
- In the creation of acoustic model clusters for unsupervised clustered-model switching we performed research in novel speaker clustering techniques. In **Chapter 6** we address the robustness problems of agglomerative hierarchical clustering (AHC) to data source variation in the context of speaker diarization. We specifically focus on the issues associated with the widely used clustering stopping method based on Bayesian information criterion (BIC) and the merging-cluster selection scheme based on generalized likelihood ratio (GLR). First, we propose a novel alternative stopping method for AHC based on information change rate (ICR). Through experiments on several meeting corpora, the proposed method is demonstrated to be more robust to data source variation than the BIC-based one. The average improvement obtained in diarization error rate (DER) by this method is 8.76% (absolute) or 35.77% (relative). We also introduce a selective AHC (SAHC), which first runs AHC with the ICR-based stopping method only on speech segments longer than 3 seconds and then classifies shorter speech segments into one of the clusters given by the initial AHC. This modified version of AHC is motivated by our analysis that the proportion of short speech segments in a data source is a significant factor contributing to the robustness problem arising in the GLR-based merging-cluster selection scheme. The additional performance improvement obtained by SAHC is 3.45% (absolute) or 14.08% (relative) in terms of averaged DER.
- In context modeling we have implemented but not at all used in evaluation due to real-time considerations an initial model of statistical models of dialog tasks created on doctor-patient interactions.
- Prominence: Detecting prominence in conversational speech. Focus on pitch accent, givenness and focus; disuency detection; sentence boundary/prominence estimation and employing acoustic and lexical correlates for improved ASR.

  With the advent of prosody annotation standards such as Tones and Break Indices (ToBI), speech technologists and linguists alike have been interested in automatically detecting prosodic events in speech. This is because the prosodic tier provides an addi-

tional layer of information over the short-term segment-level features and lexical representation of an utterance. As the prosody of an utterance is closely tied to its syntactic and semantic content in addition to its lexical content, knowledge of the prosodic events within and across utterances can assist spoken language applications such as automatic speech recognition and translation. On the other hand, corpora annotated with prosodic events are useful for building natural-sounding speech synthesizers. In **Chapter 7**, we build an automatic detector and classifier for prosodic events in American English, based on their acoustic, lexical, and syntactic correlates. Following previous work in this area, we focus on accent (prominence, or "stress") and prosodic phrase boundary detection at the syllable level. Our experiments achieved a performance rate of 86.75% agreement on the accent detection task, and 91.61% agreement on the phrase boundary detection task on the Boston University Radio News Corpus. These figures are among the best reported so far in the prosody recognition literature.

In **Chapter 8** we describe a maximum entropy based automatic prosody labeling framework that exploits both language and speech information. We apply the proposed framework to both prominence and phrase structure detection within the ToBI annotation scheme. Our framework utilizes novel syntactic features in the form of supertags and a quantized acoustic-prosodic feature representation that is similar to linear parameterizations of the prosodic contour. The proposed model is trained discriminatively and is robust in the selection of appropriate features for the task of prosody detection. The proposed maximum entropy acoustic-syntactic model achieves pitch accent and boundary tone detection accuracies of 86.0% and 93.1% on the Boston University Radio News corpus, and, 79.8% and 90.3% on the Boston Directions corpus. The phrase structure detection through prosodic break index labeling provides accuracies of 84% and 87% on the two corpora, respectively. The reported results are significantly better than previously reported results and demonstrate the strength of maximum entropy model in jointly modeling simple lexical, syntactic and acoustic features for automatic prosody labeling.

Data processing In data processing we have

- used our data selection techniques to provide DARPA and it's transcription contractor (APPEN) with the most useful data out of the Iraqi Arabic collection for translation into Farsi.
- Performed Targeted Scenario Collections. We have collected data in Farsi-Farsi interactions in the scenarios that we developed for DARPA. Despite the fact that we developed a larger set of scenarios, we employed only the ones that were in agreement with the list of topics provided by NIST.
- Automatic diacritization of Arabic scripts for automated speech processing. We have performed significant efforts in diacritizing and normalizing the transcribed data. Our efforts are continuing in identifying the potential gains achievable from the colloquial to formal transformation process.

In addition to the highlights given in this report, over 30 papers were published on related research in the past quarter and several have been accepted and are awaiting publication. The complete list of publications may be found at http://sail.usc.edu.

Furthermore, the presentation from the Nov. 2007 PI meeting is given in **Appendix B**.

Chapter 2

Rapid domain and language porting

Exploiting existing resources for rapid language modeling. Domain and language specific data selection from unstructured and noisy sources, such as the www.

There is a growing interest in using the World Wide Web (WWW) as a corpus for training models for natural language processing (NLP) tasks (5; 6; 7). One common component of many statistical NLP systems which can benefit from the use of web as a corpus is the n-gram language model. The n-gram model provides an estimate of the probability of a word sequence under Markovian assumptions. In speech recognition applications, the n-gram model is frequently used to provide a prior for decoding the acoustic sequence. The n-gram model is trained from counts of word sequences seen in a corpus and hence its quality depends on the amount of training data as well as the degree to which the training statistics represent the target application.

Text harvested from the web and other large text collections such as the English Gigaword (8) corpus provides a good resource to supplement the in-domain data for a variety of applications (9; 10). However even with the best queries and text collection schemes, both the style and content of the data acquired tend to differ significantly from the specific nature of the domain of interest. For example, a speech recognition system for spoken dialog applications requires conversational style text for the underlying language models whereas most of the data on the web is written style. To benefit from a generic corpora, we need to identify subsets of text relevant to the target application. In most cases we have a set of in-domain example sentences available to us which can be used in a semi-supervised (11; 12) fashion to identify the text relevant to the application of interest. The dominant theme in recent research literature for achieving this is the use of various rank-and-select schemes for identifying sentences from the large generic collection which match the in-domain data (9; 10). The central idea behind these schemes is to rank order sentences in terms of their match to the seed in-domain set and then select top sentences. Rank-and-select filtering schemes select individual sentences on the merit of their match to the in-domain model. As a result, even though individual sentences might be good in-domain examples, the overall distribution of the selected set is biased towards the high probability regions of the distribution.

In this chapter we build on our work in (13) and present an improved incremental selection algorithm which compares the distribution of the selected set and the in-domain examples by using a relative entropy (R.E.) criterion at each step. Section 2.1 presents several methods for data selection against which the proposed scheme is benchmarked. The proposed algorithm is described in Section 2.2. A brief description of the setup used to build the large corpus used in our experiments and other implementation details is given in Section 2.3. To validate our approach, we

present and compare the performance gains achieved by the proposed approach on two Automatic Speech Recognition (ASR) systems. The first system is a medium vocabulary system for doctor-patient conversations in English (14). The second system is a large vocabulary transcription system for European parliamentary speeches (15). Experimental results are provided in Section 2.4. We conclude with a summary of this work and directions for future research.

2.1 Rank-and-select methods for text filtering

In recent literature, the central idea behind text data selection schemes for using generic corpora to build language models, has been to use a scoring function that measures the similarity of each observed sentence in the corpus to the domain of interest (in-domain) and assign an appropriate score. The subsequent step is to set a threshold in terms of this score or the number of top scoring sentences, usually done on a heldout data set, and use this threshold as a criterion in the data selection process. A dominant choice for a scoring function is in-domain model perplexity (9; 16) and variants involving comparison to a generic language model (17; 18). A modified version of the BLEU metric which measures sentence similarity in machine translation has been proposed by Sarikaya (10) as a scoring function. Instead of explicit ranking and thresholding, it is also possible to design a classifier to Learn from Positive and Unlabeled examples (LPU) (19). In LPU, a binary classifier is trained using a subset of the unlabeled set as the negative or noise set and the in-domain data as the positive set. The binary classifier is then used to relabel the sentences in the corpus. The classifier can then be iteratively refined by using a better and larger subset of the sentences labeled in each iteration. For text classification, SVM based classifiers are shown to give good classification performance with LPU (19).

Ranking based selection has some inherent shortcomings. Rank ordering schemes select sentences on individual merit. Since the merit is evaluated in terms of the match to in-domain data, there is a natural bias towards selecting sentences which already have a high probability in the in-domain text. Adapting models on such data has the tendency to skew the distribution towards regions in the in-domain data that are highly probable. An illustration of this is short sentences containing the word 'okay' such as 'okay','yes okay', 'okay okay' which were very frequent in the in-domain data for the doctor-patient interaction task. Perplexity and other similarity measures assign a high score to all such examples, boosting the probability of these words even further. In contrast, other pertinent sentences seen rarely in the in-domain data such as 'Can you stand up please?' receive a low rank and are more likely to be rejected. Simulation results provided in (13) show the skew towards high probability regions clearly.

2.2 Data selection using relative entropy

In order to achieve an unbiased selection of data, we proposed an iterative text selection algorithm based on relative entropy (13). The idea is to select a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution.

Stolcke (20) introduced relative entropy as a measure for pruning back-off n-gram language models. In relative entropy based pruning of n-gram language models, a pruning threshold is set for the relative entropy between the n-gram distribution with history $w_1...w_{n-1}$ and the back-off n-gram distribution with history $w_2...w_{n-1}$. Higher order n-grams which have low relative entropy with respect to the lower order back-off n-grams are discarded. In this chapter, we provide a data selection algorithm based on R.E. minimization that serves a complimentary goal to R.E. based n-gram model pruning. The data selection algorithm aims at finding a good subset of data for

building language models while the goal of R.E. based pruning is to find a compact n-gram model which closely matches the unpruned model.

2.2.1 The Core Algorithm

In this section, we derive the proposed R.E. based data selection algorithm when the in-domain data is modeled using an unigram LM. A detailed derivation for the general case, when the in-domain data is modeled using a back-off n-gram language model is included in the Appendix. Let us define the following symbols:

w: word
V: Vocabulary of the in-domain model
$P(w)$: The language model built from the in-domain data[1]
$C(w)$: The count of word w in the already selected text
$N = \sum_{w \in V} C(w)$:The total number of words in the text already selected.
$c(w)$: The count of word w in the sentence being considered for selection
$n = \sum_{w \in V} c(w)$: The number of words in the sentence

The skew divergence (21) of the maximum likelihood estimate of the language model of the selected sentences to the initial model $P(w)$ is given by

$$\begin{aligned} D &= \sum_{w \in V} P(w) \ln \frac{P(w)}{(1-\alpha)P(w) + \alpha C(w)/N} \\ &= \sum_{w \in V} P(w) \ln \frac{P(w)}{\beta P(w) + \alpha C(w)/N} \end{aligned}$$

where $\beta = 1 - \alpha$.

The skew divergence is a smoothed version of the Kullback-Leibler (KL) distance with the alpha parameter denoting the smoothing influence of model $P(w)$ on the current Maximum Likelihood (ML) model. When $\alpha = 1$, the skew divergence expression is equivalent to the KL distance. Using skew divergence in place of the KL distance was useful in improving the data selection especially in the initial iterations where the counts $C(w)$ are low and the ML estimate $C(w)/N$ changes rapidly. If a sentence is selected to be included in the language model, the updated divergence is given by

$$D^+ = \sum_{w \in V} P(w) \ln \frac{P(w)}{\beta P(w) + \alpha (C(w) + c(w))/(N+n)} \tag{2.1}$$

If a sentence is not selected, then the model parameters and the divergence measure remain unchanged.

Direct computation of divergence using the above expressions for every sentence in a large corpus has a high computational cost since $O(V)$ computations per sentence are required. The number of sentences can be very large, easily on the order 10^8 to 10^9, which makes the total computation cost for even moderate vocabularies (approximately 10^5) large.

[1]The in-domain model $P(w)$ is usually represented by a linear interpolation of n-gram LMs built from different in-domain text corpora available for the task.

However given the fact that $c(w)$ is sparse, we can split the summation D^+ into

$$
\begin{aligned}
D^+ &= \sum_{w \in V} P(w) \ln P(w) \qquad (2.2)\\
&\quad - \sum_{w \in V} P(w) \ln \left(\beta P(w) + \frac{\alpha\big(C(w) + c(w)\big)}{N+n} \right) \\
&= D - \sum_{w \in V} P(w) \ln N \\
&\quad + \sum_{w \in V} P(w) \ln \big(\beta P(w) N + \alpha C(w) \big) \\
&\quad - \sum_{w \in V} P(w) \ln \left(\beta P(w) + \frac{\alpha\big(C(w) + c(w)\big)}{N+n} \right) \\
&= D + \underbrace{\ln \frac{(N+n)}{N}}_{T_1} \qquad (2.3)\\
&\quad - \underbrace{\sum_{w \in V, c(w) \neq 0} P(w) \ln \frac{\beta P(w)(N+n) + \alpha\big(C(w) + c(w)\big)}{\beta P(w) N + \alpha C(w)}}_{T_2} \\
&\quad - \underbrace{\sum_{w \in V, c(w) = 0} P(w) \ln \frac{\beta P(w)(N+n) + \alpha C(w)}{\beta P(w) N + \alpha C(w)}}_{\approx 0}
\end{aligned}
$$

Intuitively, the term T_1 accounts for the scaling of the ML probability estimates when the denominator in the estimate $C(w)/N$ increases from N to $N+n$ for all words w in the vocabulary. The term T_2 accounts for the increase in probability for words seen in the sentence where the numerator in the ML estimate increases from $C(w)$ to $C(w) + c(w)$. Equation (2.2) makes the computation of the stepwise changes in divergence tractable by reducing the required computations to the number of words in a sentence n, instead of a summation over all the words in the vocabulary i.e. $|V|$ computations. The approximation in Equation (2.3) is valid if the number of total words selected is significantly larger than the number of words expected to be seen in a single sentence ($N >> n$). As we describe in the next subsection on initialization, in the beginning of the data selection process, the counts $C(w)$ are initialized in a manner such that $N >> n$. As the data selection process selects more data, N increases reducing the approximation error further.

2.2.2 Initialization

We use the following bootstrap strategy for initializing the counts $C(w)$.

- Choose a random subset (without replacement) of the adaptation data. The size of the random subset is taken to be the same as the size of the in-domain set.
- Initialize $C(w)$ with the count of word w in the random subset. The counts are incremented by 1 to ensure non zero $C(w)$.
- The counts initialized in the previous step are used to select data using the alpha skew divergence criterion presented above.

- $C(w)$ is set to the count of the word w in the selected set. The counts are incremented by 1 to ensure non zero $C(w)$.

$C(w)$ should be non zero for ensuring finite value of T_2. In general, we have observed that in comparison to uniform initialization or initialization from a random subset, we are able to reduce the size of the selected data set by 10-15% using the two step initialization technique with no loss in performance either in perplexity or WER.

2.2.3 Alpha parameter

The alpha parameter in Equation (2.2) controls the smoothing influence of the in-domain language model. The motivation behind this smoothing was to make the relative entropy function behave smoothly during the initial part of data selection. For this purpose, a high value of alpha in the range $0.95 - 1$ was found to give good results on the two tasks described in this chapter (Section 2.4). The performance of the algorithm was not sensitive to the choice of alpha in this range. In general, a low value of alpha reduces the number of sentences selected (When $\alpha = 0$, no sentence will be selected).

2.2.4 Randomization and multiple passes

The proposed algorithm is sequential and greedy in nature and can benefit from randomization of the order in which the corpus is scanned. We generate random permutations of the sentences and select the union of the set of sentences identified for selection in each permutation. Sentences that have already been included in more than two permutations are skipped during the selection process, thus forcing the selection of different sets of sentences. After each permutation and data selection iteration, we build a language model from the union of the data selected and compute perplexity on the heldout data set. The heldout set perplexity is used as a stopping criterion to fix the number of permutations. If the perplexity increases on addition of data selected after a random permutation, no further permutations are carried out. For the purpose of fixing the number of random permutes in our experiments, we used a trigram language model with the same vocabulary as the in-domain model.

2.2.5 Smoothing

Smoothing (22) can be used after a certain fixed number of sentences are selected to modify the counts of the selected text $C(w)$. We have experimentally found out that Good-Turing smoothing after selection of every 500K words is sufficient for the tasks considered in this chapter. The impact of smoothing was not seen to be significant to warrant further exploration.

2.2.6 Extension to n-gram models

As mentioned earlier, we have introduced the data selection algorithm using unigram models to represent the in-domain data set. The extension of this R.E. based data selection algorithm to a more general, back-off n-gram model is presented in the Appendix. The computation time of the algorithm depends on the order of the n-gram model used in the data selection procedure. The number of computations required grows linearly with the total number of n-grams in the language model. In general, the total number of n-grams grows exponentially with the order of the model, making the computational cost an exponential function of the language model order (Section 2.4). For initialization of the back-off n-gram based data selection algorithm, we use a random subset of the data selected using a unigram model.

Finally, the selected data is then merged with the in-domain data set to build a language model. The choice of the order and vocabulary of this language model can be different from the order, vocabulary and choice of smoothing method used in the model that was used in the data selection procedure.

In the next section, we describe our infrastructure for collecting the large corpus used in our experiments (Section 2.4) and cover key implementation details of the proposed algorithm.

2.3 Implementation details and data collection

The vast text resources available over the world-wide web were crawled to build the large text corpora used in our experiments. Queries for downloading relevant data from the web were generated using a technique similar to (9; 17). An in-domain language model was first generated using the training material and compared to a generic background model of English text (17) to identify the terms which would be useful for querying the web. For every n-gram n in the language model we calculated the weighted ratio $p(n) \ln \frac{p(n)}{q(n)}$ where p is the in-domain model and q is the background model. The top scoring trigrams, bigrams and unigrams were selected as query terms in that order. The set of URLs returned by our search were downloaded and the non-text files were deleted. The HTML files were converted to text by stripping off tags. The converted text typically does not have well defined sentence boundaries. We found that using an off-the-shelf maximum entropy based sentence boundary detector[2] (23), seemed to improve sentence boundaries. Sentences and documents with high OOV rates were rejected as noise to keep the converted text clean.

As a pre-filtering step, we computed the perplexity of the downloaded documents with the in-domain model and rejected text which had very high perplexity (17). The goal of the pre-filtering step is to remove artifacts such as advertisements, and other spurious text. Most of these artifacts show up very clearly as a very high perplexity cluster compared to the rest of the data. Thus, by using a perplexity histogram we could easily choose and use a perplexity threshold for pre-filtering. Data were mined separately for the two ASR tasks presented in this chapter. In both cases, the initial size of the data downloaded from the web was around 750M words. After filtering and normalization the downloaded data amounted to about 500M words.

The in-domain language model against which the relative entropy of the selected set can be compared iteratively was selected in the following manner. The generalized algorithm for data selection using n-gram models (See Appendix) is significantly slower than the unigram implementation (Section 2.4) because of the need to update lower order back-off weights. Simulation experiments (13) and experiments on web-data indicate that bigram and unigram language models seem to perform well for data selection using the R.E. minimization algorithm. No performance gains were observed when using trigram models for selection. For this reason the experimental results presented in this chapter are restricted to the use of bigram models for data selection. Note however that the order of the LM used for data selection does not put any restrictions on the order of the language models used for generating query terms or the adapted language model we build from the selected data.

2.4 Experiments

To provide a more general picture of the performance of our data selection algorithm we provide experimental results on two systems which differ significantly in their system design and the nature of the ASR task that they address.

[2]MXTERMINATOR:www.id.cbs.dk/ dh/corpus/tools/MXTERMINATOR.html

The first set of experiments were conducted on the English ASR of the Transonics (14) English-Persian speech to speech translation system for doctor-patient interactions developed at USC. The second set of experiments were conducted using IBM's speech recognition system for English, submitted to the 2006 evaluation within the TC-STAR project. TC-STAR (Technology and Corpora for Speech to Speech Translation) project financed by the European Commission within the Sixth Framework Program is a long-term effort to advance research in speech to speech translation technologies[3]. The 2006 Evaluation was open to external participants as well as the TC-STAR partner sites (24).

We begin by presenting results on the Transonics task. This task was also used to provide comparisons against the large class of rank-and-select schemes described in Section 2.1. We will then provide results on the TC-STAR task. As stated in Section 2.3 bigram models generated from in-domain data were used for data selection. All language models used for decoding and perplexity measurements are trigram models estimated using Kneser-Ney smoothing.

2.4.1 Medium vocabulary ASR experiments on Transonics

The English ASR component of the Transonics speech to speech translation system is a medium vocabulary speech recognizer built using the SONIC (25) engine. We had 50K in-domain sentences (200K words) for this task to train the language model. A generic conversational-speech language model was built from the WSJ (26), Fisher (27) and SWB (28) corpora interpolated with a conversation speech LM from CMU for broadcast news (29). All language models built from the selected data and the in-domain data were interpolated with this generic conversational language model. The linear interpolation weight was determined using a heldout set. The test set for word error rate evaluation consisted of 520 utterances. A separate test set used for perplexity evaluations consisted of 5000 sentences (35K words) and the heldout set had 2000 sentences (12K words).

We report results with increasing amounts of in-domain training material, ranging from 10K sentence to 40K sentences. For every choice of in-domain training data size, we carry out data selection using the baseline methods and the proposed R.E. based method. The language models used for data selection with the perplexity rank-and-select baseline (Section 2.1) and R.E. based data selection (Section 2.2) are also built separately for every set of experiments conducted with a different in-domain data size.

We first compare our proposed algorithm against the baseline rank-and-select data selection schemes enumerated in Section 2.1. LPU and BLEU based rank-and-select schemes are computationally intensive. LPU requires iterative retraining of a binary SVM classifier which has high computational complexity. The computational complexity of the BLEU based ranking scheme is square in the order of the number of sentences. Our results which include comparisons against these two systems are thus limited to a smaller 150M word web-collection. The thresholds for data selection using the ranking based baselines were fixed using the heldout set perplexity.

For the 150M word web-collection, Table 2.1 shows the fraction of sentences selected and the resulting perplexity and WERs for the various data selection schemes with different amounts of in-domain data used to seed the data selection. In Table 2.1, *NoWeb* refers to the language model built solely from in-domain data and *AllWeb* refers to the case where the entire 150M web-collection was used. As the comparison shows, the proposed algorithm outperforms the rank-and-select schemes with just 10% of data selected from the web collection. The best reduction in perplexity with the proposed scheme is 5% relative corresponding to a reduction in WER of 3% (relative).

One of the goals of the Transonics task was to find an optimal vocabulary size as the initially available data was quite small (30). Hence the vocabularies of the language models used to compute

[3]Project No. FP6-506738

		Number of in-domain sentences		
		10K	20K	40K
Perplexity	NoWeb	60.0	49.6	39.7
	AllWeb	57.1	48.1	38.2
	PPL	56.1	48.1	38.2
	BLEU	56.3	48.2	38.3
	LPU	56.3	48.2	38.3
	Proposed	**53.7**	**46.6**	**38.0**
Word error rate (in %)	NoWeb	19.8	18.9	17.9
	AllWeb	19.5	19.1	17.9
	PPL	19.2	18.8	17.9
	BLEU	19.3	18.8	17.9
	LPU	19.2	18.8	17.8
	Proposed	**18.1**	**17.9**	**17.1**
Data selected (in %)	NoWeb	0	0	0
	AllWeb	100	100	100
	PPL	93	92	91
	BLEU	91	90	89
	LPU	90	88	87
	Proposed	**11**	**10**	**11**

Table 2.1: Transonics: Perplexity, Word error rate and percentage data selected for different number of initial sentences for a corpus size of 150M

perplexity presented in Table 2.1 are different. However the OOV rate on the heldout data was less than 1% for all the vocabularies. The vocabularies for the language models in Table 2.1 ranged from 70K to 110K words.

To get a more complete picture of the relationship between performance and amount of data selected, we also conducted experiments using simulations (13) where we restricted the number of sentences selected by the perplexity ranking baseline to be the same as the number of sentences selected by the proposed method. For these simulations, we generated samples from a reference language model using a random walk procedure[4]. We then compared the performance of data selection using perplexity-based ranking and the R.E. criterion with two metrics. The first metric computes perplexity on a heldout set and the second computes the the relative entropy with respect to the reference model (32). In both these metrics the R.E. based criterion outscored perplexity-based selection. In fact, for many cases selecting a random subset of data was found to give better performance using both metrics when compared to the baseline perplexity-based ranking method (13).

Table 2.1 presented results on data selection from a corpus of 150M words. In order to study the performance of this algorithm on larger corpora, we used a larger data set of 850M words which consisted of the medical domain collection of 320M words collected from the web and a 525M word collection published by the University of Washington for the Fisher corpus (33; 34).

We provide comparisons with only the perplexity based rank-and-select scheme, as the LPU and BLEU based schemes do not scale well to large text collections. More importantly, our results on the 150M word corpus (See Table 2.1) suggest that the performance of the ASR system is

[4]In the SRILM (31) tool kit, a random sample can be generated by ngram -gen

		Number of in-domain sentences		
		10K	20K	40K
Perplexity	NoWeb	60.0	49.6	39.7
	AllWeb	56.9	47.7	38.2
	PPL	55.8	47.4	38.2
	Proposed	**52.1**	**45.2**	**36.8**
Word error rate (in %)	NoWeb	19.8	18.9	17.9
	AllWeb	19.3	19.1	17.9
	PPL	19.1	18.7	17.9
	Proposed	**17.8**	**17.6**	**17.0**
Data selected (in %)	NoWeb	0	0	0
	AllWeb	100	100	100
	PPL	88.5	87.8	87.3
	Proposed	**9.3**	**10**	**8.7**

Table 2.2: Perplexity, Word error rate and percentage data selected for different number of initial sentences for a corpus size of 850M

	unigram	bigram	trigram
AllWeb	105K	25.3M	36.2M
PPL	99K	22.1M	32.4M
Proposed	**70K**	**3.2M**	**8.2M**

Table 2.3: Transonics: Number of estimated n-grams with web adapted models for different number of initial sentences for the case with 40K in-domain sentences. Corpus size=850M

approximately the same when using data selected from any one of the LPU, BLEU, or PPL based data selection schemes.

The results on the 850M word set, measured in terms of PPL and WER (Table 2.2) follow the same trend as in the 150M data set. The importance of proper data selection is highlighted by the fact that there was little to no improvement in the unfiltered case (*AllWeb*) by adding the extra data as is, whereas consistent improvements can be seen when the proposed iterative selection algorithm was used. Perplexity reduction in relative terms was 7%, 5% and 4% for the 10K, 20K and 40K in-domain set, respectively. Corresponding WER improvements in relative terms were 6%, 4% and 4% respectively. Table 2.2 also shows that the amount of data selected by the R.E. based data selection scheme was a factor of 9 smaller than the entire collection of 850M words and still provided improved ASR performance.

It is interesting to note that for our Transonics experiments, the perplexity improvements correlate surprisingly well with WER improvements. This is in contrast to previous studies in speech recognition (35) where WER improvements did not correlate well with perplexity.

Our results on the medium-vocabulary Transonics task, indicate that with the proposed scheme, we can identify significantly smaller sets of sentences such that the models built from the selected data have a substantially sparser representation and yet perform better (in terms of both perplexity and WER) than models built from the entire corpus. Next, we present results on a large vocabulary task.

Fraction of data selected(words)	Baseline	All (525M)	1/11 (45M)	1/7 (71M)	1/3 (170M)
Perplexity(Dev)	115	94.5	94.5	91.3	88.7
WER (Dev)%	11	10.7	10.9	10.8	10.6
WER (Eval)%	8.9	8.4	8.6	8.5	8.5

Table 2.4: TC-STAR: Performance comparison of the language models built with different fractions of data being selected for the Dev06 and Eval06 test sets. The baseline had 525M words of fisher web data (1) (U.Wash) and 204M words of Broadcast News (2) (BN) as out-of-domain data. The WER on Dev06 for the baseline was 11% and 8.9% on Eval06

2.4.2 Large vocabulary experiments on TC-STAR

To contrast with the medium vocabulary single decoder Transonics system, we conducted experiments on the IBM LVCSR system used for transcription of European Parliamentary Plenary Speeches (EPPS) (15) as part of the TC-STAR project. We present results on two test sets, namely, the development (Dev06) and evaluation (Eval06) test sets. The Dev06 test set consists of 3 hours of data from 42 speakers (mostly non-native speakers). The Eval06 test set comprises of 3 hours of data from 41 speakers. The Dev06 and Eval06 test sets cover parliamentary sessions between June and September 2005 and contain approximately 30K words each.

The baseline system's interpolated language model was built from two in-domain EPPS data sources, namely, the transcripts used for training acoustic models (2M words) and the Final Text Editions (FTE) (33M words) and two out-of-domain data sources, the University of Washington's Fisher web data corpus(525M words) and data from the Broadcast News domain (204M words). The baseline performance of the best ASR system on the Dev06 test set was 11% and, 8.9% on the Eval06 test set. We provide performance comparisons against this baseline by replacing the two out-of-domain data sources in the baseline system with increasing fractions of text selected by the R.E. based data selection method. As can be seen from Table 2.4, incorporating the 525M words mined by our crawling scheme (Section 2.3) boosted the system performance to 8.4% (6% relative over the baseline). The effectiveness of the data selection scheme is demonstrated by the fact that similar performance gains over the baseline are obtained (8.5% and 8.4% WER) when using $1/7^{th}$ of the data i.e., 70M words or all the data i.e., 525M words. When the data selected is increased to $1/3^{rd}$ of the total size i.e., 170M words, the WER reduction is similar to that seen earlier despite a modest reduction in perplexity. A further reduction in WER was achieved when a third out-of-domain source, the 204M word Broadcast News corpus was included:

- $1/7^{th}$ of the selected data yielded a reduction of WER from 11% to 10.6% and 8.9% to 8.3% on the Dev06 and Eval06 test sets, respectively.
- $1/3^{rd}$ of the selected data yielded a reduction of WER from 11% to 10.3% and 8.9% to 8.3% on the Dev06 and Eval06, respectively.

The ASR system used for transcribing English EPPS speeches in the TC-STAR 2006 evaluation used a system combination approach, the detailed architecture is described in (15). The best performance was obtained with a system combination using ROVER (36), i.e., 10.4% and 8.3% on the Dev06 and Eval06 test sets. The equivalent system combination result when using the R.E. based data selection scheme that selected $1/3^{rd}$ of the data yielded 9.8% and 7.9% WER on the Dev06 and Eval06 test sets respectively, a significant improvement in performance at these low levels of WER. To further understand the significance of the data selected using the proposed scheme, we selected $1/3^{rd}$ of the data from the same corpus randomly and studied the performance

Task/n-gram order	unigram	bigram	trigram	4gram	5gram
TC-STAR	1.0	5.2	22.3	117.0	560.2
Transonics	1.0	3.6	13.0	44.1	180.1

Table 2.5: TC-STAR : Time required for data selection with increasing order of the data selection language model normalized by the time required with unigram language model

of the ASR system. This yielded a WER of 10.8 % on the Dev06 and 8.6% on the Eval06 test sets thereby indicating that the proposed scheme does indeed select data that helps in improving ASR performance in terms of WER.

2.4.3 Computation time and n-gram order

The computation time of the proposed R.E. based data selection algorithm depends on the order of the n-gram language model used for data selection (See Appendix). In our experiments, we observed an exponential trend in computation time with increasing n-gram order. Table 2.5 shows the computation time required with higher order language models normalized by the computation time for a unigram model. A detailed theoretical and experimental analysis of the interplay between the language model order, number of parameters and the computation time has not been carried out at this stage. We intend to undertake this analysis in our future work on data selection.

2.5 Discussion and analysis of results

It is interesting to compare the data selection results between the Transonics and TC-STAR experiments. For Transonics, we used a web corpus of 320M words (excluding Fisher data). The data selection algorithm was able to achieve better performance than the out-of-domain LM built from the entire 320M word corpus, while selecting just $1/10^{th}$ of the data. In contrast the IBM TC-STAR system requires significantly more data. However, if we consider the ratio of the selected data size with in-domain training data size we find the results much more comparable. This is expected since with good in-domain training data the dependency on out of domain data is less. In addition, the Transonics ASR system had a higher baseline WER than the TC-STAR system.

More insights into these results can be gained by comparisons with the performance of the ROVER-based TC-STAR system. First, the 525M word collection generated using the scheme presented here gave an improvement of 0.5% compared to the baseline which used two out-of-domain sources of over 700M words. The baseline WER can be achieved with just 70M words selected from an out-of-domain source (instead of 700M words). Second, careful data selection can yield the same gains as those obtained from a system combination approach.

2.6 Conclusion

2.6.1 Summary of Contributions

In this chapter we presented a novel scheme for selecting *relevant* subsets of sentences from large collections of text acquired from the web. Our results indicate that with this scheme, we can identify significantly smaller sets of sentences such that the models built from the selected data have a substantially sparser representation and yet perform better (in terms of both perplexity and

WER) than models built from the entire corpus. On our medical domain task which had sparse in-domain data (200K words), we were able to achieve around 4% relative improvement in WER with a factor of 7 reduction in language model parameters while selecting a set of sentences $1/10^{th}$ the size of the original corpus. For the TC-STAR task where the in-domain resources were much larger (50M words), we achieved 6% relative WER improvement by using just $1/3^{rd}$ of the data. Although most of our results in this chapter were on data acquired from the web, the proposed method can easily be used for adaptation of domain specific models from other large generic corpora.

2.6.2 Scope of this work

The research effort presented in this chapter is directed towards selecting relevant domain specific data from large collections of generic text. We make no assumptions on how the data were collected or what specific web crawling and querying techniques are used. The methods we have developed can be seen as supplementing the research efforts by the machine translation community on identifying web resources (7; 37) or using web counts (6) for language modeling. We also believe that this work can augment topic based LM adaptation techniques. Topic based LM adaptation schemes typically use LSA (38) or variants (39) to automatically split the available training text across multiple topics. This allows for better modeling of each individual topic in the in-domain collection. The trade off is that since the available text is split across topics, each individual model is trained on less data. We believe that this problem can be addressed by selecting data for each topic from a large generic corpora using the proposed data selection algorithm.

2.6.3 Directions for future work

The effect of varying data granularity has not been studied in this work. We have used sentence level selection, but the selection process can also be naturally extended to groups of sentences, fixed number of words, paragraphs or even entire documents. Selection of data in smaller chunks has the potential to select data better suited to the task but may result in over-fitting to the existing in-domain distribution. In such a case the adaptation model will provide little extra information to the existing model. We plan to study the effect of this trade-off between data novelty and match to in-domain model on the LM performance, for different levels of selection granularity. We are also looking into extending the algorithm to work directly on collections of n-gram counts. One motivation for research in this direction is that Google has released aggregate unigram to 5-gram counts for their web snapshot (40).

The proposed method can be combined with rank-and-select schemes described in Section 2.1. We are exploring the use of ranking to reorder the data such that the sequential selection process gives better results with fewer number of randomized searches.

The current framework relies on multiple traversals of data in random sequences to identify the relevant subset. An online single-pass version of the algorithm would be of interest in cases where the text data is available as a continuous stream (one such source is RSS feeds from blogs and news sites). If updates from the stream sources are frequent, iterating through the entire text collection is not feasible. Some ideas we are investigating to make the selection process single-pass is to use multiple instances of the algorithm with different initial in-domain models generated by bagging. Voting across these multiple instances can be then used to select data. Finally we are also investigating how to select sentences with a probability proportional to the relative entropy gain instead of the threshold based approach currently being used.

Chapter 3

Features for Robust ASR

Early Auditory Processing Inspired Features for Robust Automatic Speech Recognition

Hearing is one of the most highly developed senses in humans. The human auditory system can robustly localize, segment, and recognize sounds embedded in complex scenes. In contrast, machine recognition performance degrades drastically in various conditions such as in the presence of noise, speaker changes or overlapping sources. Despite years of intensive research in speech production and psychoacoustic analysis of human auditory system, the machine speech and audio processing methods still remain poor cousins to their biological counterparts. Understanding and modelling the information processing architectures in biological systems can offer the possibility of reducing the performance gap between human and machines in realistic conditions.

In the literature, there are signal representation methods based on physiological evidence such as linear predictive coding (LPC) and Mel-frequency cepstral coefficients (MFCC). While the LPC is related to the speech production model, the MFCC is based on a crude approximation of critical bands in the human auditory system. These features have been successfully used in speech recognition, audio classification, and auditory scene analysis, however, they are highly susceptible to noise. The perceptually inspired method called RelAtive SpecTrAl (RASTA) processing has been shown to improve robustness of speech recognition in the presence of noise (41). It includes critical band analysis, temporal filtering, and equal loudness adjustment. It is designed to remove noise components by filtering out the slowly changing or steady-state factors interfering with the speech source.

There has also been research in the area of computational modelling of early and central stages of human auditory system for audio and speech processing. For example in (4; 42), it has been shown that their proposed early auditory model is robust to noise. This was used in (43) to extract MFCC-equivalent features by sampling the output of auditory spectrum at the channels corresponding to the MFCC's critical bands. In (44), robust processing was proposed by combining MFCC-type front end with an auditory based model. However, the speech recognition performance with the proposed feature set was not superior to MFCC (43; 44). On the other hand, it has also been shown that multi scale spatio-temporal modulation features derived from central stages of auditory system are robust to noise in a classification task in (45), but these features are computationally very expensive for downstream processing since they produce a large dimensional tensor representation.

In this chapter, we present biologically inspired robust speech processing algorithms based on human auditory system. As mentioned before, the multi-scale cortical representation of central auditory system is computationally very expensive (43; 45). Hence we focus only on the early

auditory (EA) processing which is computationally less expensive and has also been shown to be robust to noise. The contributions of this work are as follows. First, we develop an auditory processing based feature by replacing the triangular filter bank in MFCC feature extraction with a model that is more faithful to the processing stages in the EA system. The EA model used here consists of cochlear filtering, inner hair cell, and lateral inhibitory stages mimicking the process from basilar membrane to the cochlear nucleus in the auditory system. Then, a novel feature extraction algorithm is proposed which retains only the cochlear channel outputs that are more likely to fire neurons in the central auditory system by using principal component analysis (PCA). We also show empirically that an additional nonlinear compression modelling the outer hair cells has significant improvement on the speech recognition performance of the extracted feature set. The robustness of the developed features to a variety of noisy scenes is tested in a speech recognition task using the Aurora 2.0 database, and compared with state of the art MFCC and RASTA features. The experimental results show that the proposed feature set is more robust to noise compared to MFCC and RASTA features.

The chapter is organized as follows. In Section 2, an overview of EA spectrum estimation along with the auditory based feature extraction is provided. In Section 3, the experimental set-up together with the preliminary speech recognition results is discussed. Section 4 presents analysis of the auditory model, and explains the robust features obtained by post processing the EA model output. The experimental results are detailed in Section 5.

3.1 Early Auditory Processing and Auditory Based Features

In the human auditory system, when acoustic signal enters the ear, sound pressure waves create vibrations along the basilar membrane of cochlea. The cochlea separates the incoming signal frequencies by responding to different frequencies in different spatial locations along its length. Hence, the basilar membrane can be thought as a bank of band-pass filters $h(t;s)$ tonotopically ordered along the length of cochlea (4). The spectral analysis performed by the cochlear filters is implemented as a bank of 128 overlapping constant-Q asymmetric band-pass filters (4). The central frequencies of the band-pass filters are uniformly distributed along a logarithmic frequency axis (s).

The inner hair cell (IHC) stage transfers cochlear filter outputs into auditory nerve patterns. The IHC stage can be modelled in three steps: a high-pass filter $u(t)$ corresponding to fluid-cilia coupling, followed by a nonlinearity $g(\cdot)$ corresponding to ionic channel, and a low-pass filter $w(t)$ to model the leakiness of hair cell membrane (4). Here, $g(\cdot)$ is implemented by a sigmoidal function, and $w(t)$ is implemented to represent phase-locking decrement in the auditory nerve beyond 2 kHz (4).

A hair cell fires when the potential builds up along the hair cell membrane. Auditory nerve fibers carry this neural spike to the cochlear nucleus of the central auditory system. In the cochlear nucleus, a lateral inhibitory network (LIN) detects discontinuities along the tonotopic axis (4).The LIN is modelled by a first-order spatial derivative (∂s) followed by a half wave rectifier (HWR) that models the nonlinearity of the neurons in the LIN. Here, the spatial derivative is approximated by a difference operation between adjacent frequency channels. The two-dimensional output, *auditory spectrum* (4), is obtained after leaky integration ($\int_T$) mimicking the inability of central neurons to follow rapid temporal changes.This stage is implemented as temporal filtering over a short time window, $\mu(t;\tau) = e^{-t/\tau}u(t)$, with time constant $\tau = 16$ ms. The block diagram of this early auditory processing is shown in Fig 3.1.

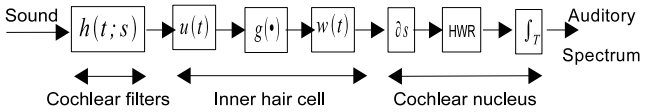


Figure 3.1: A computational model of processing in the early auditory system (4)

The widely used MFCC is based only on a crude approximation of basilar membrane filtering in the cochlea, and it has been shown empirically that it is highly susceptible to noise. Using a more accurate model of the auditory system can essentially help to obtain a better performance compared to the MFFC under noisy conditions. For this purpose, we introduce auditory based features (ABF) by replacing the triangular filter bank analysis stage used in MFCC computation with aforementioned early auditory processing model. The ABF is expected to be robust to noise due to LIN and IHC stages in the EA model. The spatial derivative used in the LIN reduces the effect of noise due to the difference operation between adjacent channels, and the phase locking activity in IHC stage enhances the signal (42). To obtain ABF, we compute the discrete cosine transform (DCT) of the logarithm of the *auditory spectrum*, and keep 13 of the coefficients as in the MFCC computation. The first (Δ) and second order time derivative ($\Delta\Delta$) features are appended to the raw features to form 39−dimensional feature vector. In all of the proposed feature extraction methods in the following sections, Δ and $\Delta\Delta$ features are used together with raw features, unless stated otherwise. The block diagram of ABF extraction is summarized in Fig. 3.2(a), where "EA Model" box represents the early auditory process shown in Fig. 3.1.

3.2 Experimental Setup and Preliminary Results

To validate the new auditory based features, we perform speech recognition task on the Aurora 2.0 database (46) using the Hidden Markov Model Toolkit (HTK) (47). The database consists of connected digits degraded with different noise conditions under different signal-to-noise ratios (SNR). We used 8440 clean utterances from 55 female and 55 male adults for training, and the recognition is done using the test sets with varying SNR levels (mis-matched training/testing). Training and testing follows the specifications detailed in (46).We created HMM word models for digits with 16 states per digit and 3 Gaussian mixtures per state. A three state silence model with 6 Gaussian mixtures per state and a one state short pause model which is tied to the middle stage of silence model are used. There are two sets of testing data; Set A and Set B. Set A contains the noise types of subway, babble, car and exhibition hall and Set B contains restaurant, street, airport , and train station noise at various SNR levels.

For MFCC feature extraction, we followed the specifications given in (46). The 39-dimensional MFCC features consisting of 13 cepstral features plus Δ and $\Delta\Delta$ are used as a baseline. 23 channels were used during MFCC computation. The frame size was 25ms, and the frame shift was 10ms. The ABF extraction details were presented in Section 3.1.

The speech recognizer performance using both MFCC and ABF features is shown in Fig 3.3. Here and in the preliminary results presented in Sec. 4.1 and 4.2, the data were degraded with subway noise for varying levels of SNR. We obtained similar results for other noise types as well (discussed later in Section 3.4 in detail). It can be observed from Fig 3.3 that replacing Mel-filterbank with a more accurate early auditory model improves the speech recognition performance under noisy conditions. These were our initial experiments to understand the potential of EA

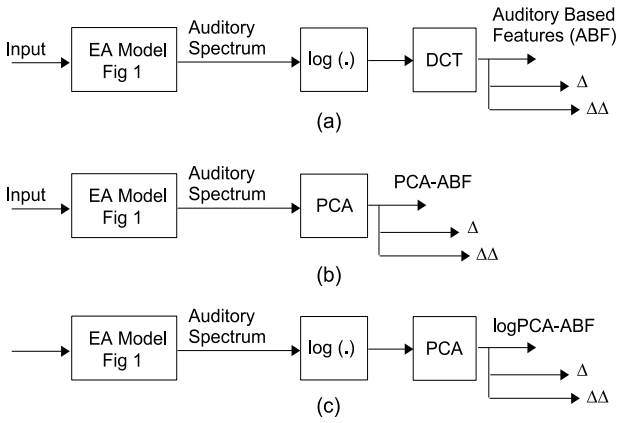


Figure 3.2: Block diagrams of feature extraction algorithms: (a) ABF feature extraction (b) PCA-ABF feature extraction (c) logPCA-ABF feature extraction

modelling, and to see the effect of using a more detailed feature model on speech recognition performance. The ABF treats all of the auditory channel outputs with equal importance. However, the channels with stronger stimulus might carry more information as explained in the next section. Thus, the auditory spectrum is post-processed before feeding it into the speech recognizer to further improve the noise robustness. The details are presented in the next section.

3.3 Post-processing of Auditory Spectrum

3.3.1 Principal Components of Auditory Spectrum

The output of the early auditory model is transferred to the neurons in the central auditory system. The final stage of the early auditory model, leaky-integration, represents a simplified model of a leaky-integrate-and-fire (LIF) neuron model. These types of neurons accumulate the charges delivered by synaptic input, generate a spike when a threshold is reached, and reset the capacitive charge to zero after spike generation (48). The stronger the stimulus, the higher is the chance of neuron getting fired.

The auditory spectrum obtained from the model presented in Fig. 3.1 represents the output of leaky integration. Here, it is assumed that the channel outputs that fire neurons carry the most significant information. Hence, we find the channel outputs that are more likely to generate a spike. Since a stronger stimulus has a better chance to generate a spike, the filter outputs are linearly transformed to a reduced dimension such that the reduced dimension features represent the strong components of the spectrum, which also means preserving the most of signal energy. To do this, we apply PCA (49) at the output of early auditory model. We retain only the most significant information by using PCA.

PCA is a dimension reduction technique that tries to obtain the best representation of the original data in the least squares sense in the projected space. Let $X = [x_1 x_2 \cdots x_N]$ be $d \times N$ data matrix, where $d = 128$ is the original data dimension and N is the sample size, and $W = [w_1 w_2 \cdots w_m]$ is the $d \times m$ transformation matrix, where $1 \leq m \leq d$. The goal of PCA is to find $\hat{W}$ such as:

$$\hat{W} = \arg\min \sum_{j=1}^{N} \|x_j - \sum_{i=1}^{m} (w_i^T x_j) w_i\|^2. \tag{3.1}$$

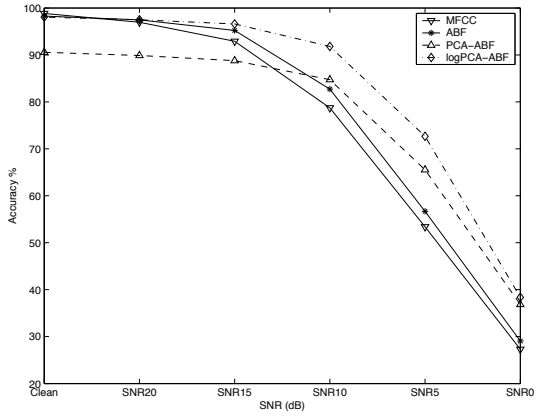


Figure 3.3: Speech recognition results with MFCC, ABF, PCA-ABF ($m = 10$), and logPCA-ABF ($m = 25$) for connected digits data with subway noise

The problem reduces to finding eigenvalues of the sample covariance matrix $S = \frac{1}{N}\sum_{k=1}^{N}(x_k - u)(x_k - u)^T$, where u is the sample mean. The columns of $\hat{W}$ called principal components are the eigenvectors that correspond to the m largest eigenvalues of the data.

To set the number of principal components, we compute

$$\alpha_m = \frac{\sum_{k=1}^{m}\lambda_k^2}{\sum_{i=1}^{d}\lambda_i^2}. \tag{3.2}$$

α_m represents the portion of signal energy retained by keeping m principal components. We set m such that α_m is larger than 0.95, and we also consider the speech recognition performance.

The new feature extraction algorithm based on principal components of auditory spectrum, named as PCA-ABF, is summarized in Fig 3.2(b). The PCA transformation matrix W is learnt using the clean training data. The number of principal components retained is varied in the automatic speech recognition (ASR) experiments. The best ASR performance is achieved with $m = 10$ and $\alpha_{10} = 99\%$. In Fig 3.3, the speech recognition results with PCA-ABF are shown together with MFCC and ABF performance. The ASR results in Fig 3.3 show that PCA-ABF outperforms both ABF and MFCC for low SNR levels, whereas it performs poorer compared to ABF and MFCC for speech with moderate or high SNR levels. The PCA might be reducing the class discrimination for clean speech since only the significant channel outputs are retained, and this can cause some information loss about the source.However for speech with low SNR level, the gain achieved with the removal of noise components with PCA is higher than the source information loss, resulting in ASR performance improvement. Fig 3.3 shows that finding the principal components of EA model output is beneficial for low SNR levels. Thus, we kept PCA in our feature extraction algorithm but with an additional compression step modelling the outer hair cells (OHC) as explained in the next section.

3.3.2 Principal Components of Compressed Auditory Spectrum

The auditory nerves have limited dynamic range (50). The dynamic range of basilar membrane and the neural response are compressed nonlinearly by the OHC. The OHC provide greater amplification

Table 3.1: Speech Recognition Results with MFCC (Accuracy %)

	Set A					Set B				
	Subway	Babble	Car	Exhibition	Avg.	Restaurant	Street	Airport	Station	Avg.
Clean	98.83	98.97	98.81	99.14	98.94	98.83	98.97	98.81	99.14	98.94
SNR20	96.96	89.96	96.84	96.2	94.99	89.19	95.77	90.07	94.38	92.35
SNR15	92.91	73.43	89.53	91.85	86.93	74.39	88.27	76.89	83.62	80.79
SNR10	78.72	49.06	66.24	75.1	67.28	52.72	66.75	53.15	59.61	58.06
SNR5	53.39	27.03	33.49	43.51	39.36	29.57	38.15	30.69	29.74	32.04
SNR0	27.3	11.73	13.27	15.98	17.07	11.7	18.68	15.84	12.25	14.62
Avg.	74.69	58.36	66.36	70.30	67.43	59.4	67.77	60.91	63.12	62.80

Table 3.2: Speech Recognition Results with ABF (Accuracy %)

	Set A					Set B				
	Subway	Babble	Car	Exhibition	Avg.	Restaurant	Street	Airport	Station	Avg.
Clean	98.26	98.44	98.35	98.66	98.43	98.26	98.44	98.35	98.66	98.43
SNR20	97.47	91.56	96.91	96.1	95.51	90.45	96.19	90.9	95.44	93.25
SNR15	95.22	78.68	91.77	92.11	89.45	77.11	90.4	83.23	88.93	84.92
SNR10	82.72	57.3	70.86	76.68	71.89	58.26	69.1	60.87	66.39	63.66
SNR5	56.69	32.81	35.74	45.99	42.81	37.5	40.25	36.61	32.78	36.79
SNR0	29.06	15.46	15.78	18.87	19.79	16.55	21.29	19.48	13.14	17.62
Avg.	76.57	62.38	68.24	71.4	69.65	63.02	69.28	64.91	65.89	65.78
RI-M	7.43	9.65	5.59	3.7	6.82	8.92	4.69	10.23	7.51	8.01

to signals at low levels. We modified our model, and used logarithmic amplitude transformation to model the nonlinear compression due to OHC, and then applied PCA to the compressed auditory spectrum. This new feature set is called logPCA-ABF. The block diagram of logPCA-ABF feature extraction is shown in Fig. 3.2(c). The best ASR performance was achieved with $m = 25$ and $\alpha_{25} = 99\%$ for logPCA-ABF features. The ASR experiment results with logPCA-ABF features are shown in Fig. 3.3 together with the other features. Fig. 3.3 shows that with logPCA-ABF the performance degradation faced with using PCA-ABF feature set for speech with moderate or high SNR level was resolved. Since the dynamic range is reduced due to the compression, we have to retain more principal components to have $\alpha_m = 0.99$. Thus, it can be expected that there is more detailed information compared to PCA-ABF method with increased number of principal components here, and this improves the results for clean speech. Also, with logPCA-ABF the ASR performance improved even more for speech with low SNR levels. The results with other noise types are discussed in Sec. 5.

3.4 Experiment Results

The details of the speech recognition task were presented in Section 3.2. We used MFCC features as a baseline to compare our speech feature representations performance with. Also, we compared the speech recognition performance of our final feature set logPCA-ABF with RASTA features.

The speech recognition word accuracy results are given in Tables 3.1-3.4. For each noise type, we computed average word accuracy (denoted as "Avg.") in the tables with the results of all SNR levels including "clean" speech. We also computed relative word error rate (WER) improvement. In Table 3.2 and 3.4, "RI-M" and "RI-R" values show the relative WER improvement over MFCC and RASTA features, respectively.

Table 3.3: Recognition Results with RASTA (Accuracy %)

	Set A					Set B				
	Subway	Babble	Car	Exhibition	Avg.	Restaurant	Street	Airport	Station	Avg.
Clean	98.7	98.93	98.99	99.1	98.93	98.7	98.93	98.99	99.1	98.93
SNR20	98.41	97.89	98.57	97.46	98.08	97.33	97.8	97.59	98.44	97.79
SNR15	96.68	93.85	95.14	95.17	95.21	94.4	94.03	94.96	94.96	94.59
SNR10	85.25	79.11	75.15	79.44	79.74	83.45	77.28	82.85	79.56	80.79
SNR5	56.14	49.01	38.2	43.61	46.74	57.48	48.01	54.48	46.52	51.62
SNR0	31.55	25.74	19.43	15.88	23.15	30.58	24.62	31.98	25.17	28.09
Avg.	77.79	74.09	70.91	71.78	73.64	76.99	73.45	76.81	73.96	75.3

The experiment results with MFCC feature set, and ABF set are given in Table 3.1 and Table 3.2, respectively. It can be observed that ABF performs better for noisy speech compared to MFCC. The average relative WER improvement obtained with ABF was 6.82% for Set A and 8.01% for Set B, resulting in overall 7.42% WER improvement over MFCC baseline. We believe that the slight performance degradation with ABF for clean speech is due to the lateral inhibitory network in the auditory model. In the LIN, while taking the difference of adjacent channels reduces the noise when the speech is noisy, this can cause information loss or introduce noise to clean speech. We can conclude from these experiments that using a better model of auditory system helps to improve speech recognizer performance when the speech is contaminated with noise.

The experiment results with RASTA and logPCA-ABF features are given in Table 3.3 and Table 3.4, respectively. The speech recognition performance of logPCA-ABF is compared with both MFCC and RASTA features. The average recognition result with Set A improved from 67.43% to 78.90%, and from 62.80% to 79.62% for Set B resulting in 35.2% and 45.2% relative WER improvement over the MFCC baseline. Similarly, with logPCA-ABF features the relative WER improvement over the RASTA feature performance was 19.94% and 17.47% for Sets A and B, respectively. It is clear that logPCA-ABF features work well for not only stationary noise types (i.e. car, exhibition hall (46)) but also non-stationary noise types (i.e. street, airport (46)). Overall, the logPCA-ABF features provide 40.2% and 18.71% relative WER improvement over the MFCC and RASTA features performance, respectively.

To compare all the results, we computed the average word accuracy over all noise types for each noise level condition, i.e. the average word accuracy for clean speech over all eight noise types. The results for all methods are presented in Fig 3.4. It is clear that the improvement gained with logPCA-ABF features is substantial, and it outperforms both MFCC and RASTA features in noisy conditions.

3.5 Conclusion and Future Work

In this chapter, we derived bio-inspired features for automatic speech recognition based on the processing stages in the early human auditory system. The derived features are validated in a speech recognition task in the presence of variety of noise types. First, we implemented an auditory based feature by replacing the Mel-filterbank analysis stage in MFCC feature extraction with an auditory model that consists of cochlear filtering, inner hair cell, and lateral inhibitory network stages. In

Table 3.4: Recognition Results with logPCA-ABF (Accuracy %)

	Set A					Set B				
	Subway	Babble	Car	Exhibition	Avg.	Restaurant	Street	Airport	Station	Avg.
Clean	98.06	98.6	98.17	98.34	98.29	98.06	98.6	98.17	98.34	98.29
SNR20	97.5	96.46	97	96.27	96.81	95.17	96.68	94.37	96.17	95.6
SNR15	96.61	93.94	95.55	95.34	95.36	92.57	95.21	93.81	94.23	93.96
SNR10	90.81	86.24	85.89	88.09	87.76	87.66	86.73	87.08	86.08	86.89
SNR5	72.66	68.01	50.92	67.58	64.79	74	65.39	67.55	59.8	66.69
SNR0	38.37	32.9	22.49	27.68	30.36	44.45	31.08	39.33	30.23	36.27
Avg.	82.34	79.36	75	78.88	78.9	81.99	78.95	80.05	77.48	79.62
RI-M	30.21	50.43	25.69	28.9	35.2	55.63	34.68	48.97	38.92	45.2
RI-R	20.46	20.33	14.07	25.17	19.94	21.71	20.71	13.98	13.5	17.47

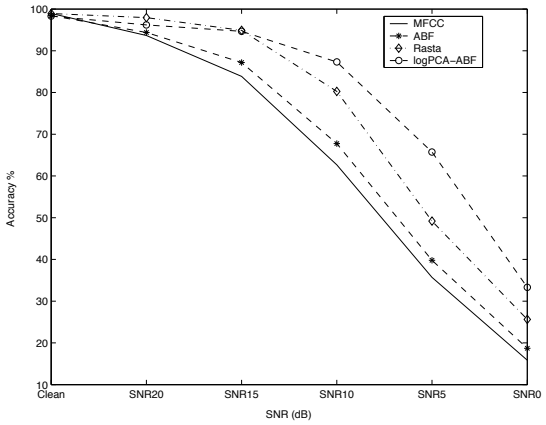


Figure 3.4: Performance comparison of all methods. Accuracy is the average of recognition results over all noise types. The proposed logPCA-ABF outperforms all other methods.

our experiments, it was shown that the ABF was more robust to noise compared to MFCC. We derived a new set of features by post-processing the early auditory spectrum. In the experiments, it was shown that the selected features of nonlinearly compressed early auditory spectrum via PCA provided substantial improvement over both MFCC and RASTA features in noisy conditions. This is attributed to the noise suppressing feature of LIN, and signal enhancement feature of IHC stages in the EA model. Also, by performing PCA on the nonlinearly compressed EA spectrum, only the channel outputs that are more likely to transmit information to the neurons in the central auditory system are selected, thereby removing insignificant channel outputs together with noise.

The experiment results showed the importance of two stages added to the early auditory model: *i*) the compression in the OHC *ii*) the selection of significant components of leaky integration taking place in the cochlear nucleus. As part of our future work, we plan to model the OHC compression more accurately as an adaptive model. We will also develop methods that can help us to code the spikes generated at the output of leaky integration such that it will represent relevant information more robustly.

Table 4.1: User model dimensions(Dimension 1,2,3) based on the knowledge about people (3)

Dim. 1	A single, canonical user	A group, collection of users
Dim. 2	Specified by the system designer	Inferred by the system
Dim. 3	Long term	Short term

Chapter 4

Multimodal User Behaviors

Analyzing the Multimodal Behaviors of Users of a Speech-to-Speech Translation Device by using Concept Matching Scores

Spoken conversations have been recognized as the primary communication mechanism between humans. With increasing globalization, the need for cross-lingual interactions has become a necessity for a variety of domains including business and travel. As speech and language technologies evolve, we can envision intelligent speech-enabled systems mediating dialogs between people who do not share a language, through automated speech to speech translation. Significant progress is being made in this direction by several research institutions (51; 52; 53; 54). The goal of such systems is to be truly cognizant of the interaction, intelligent and performing as a communication aide, beyond serving as a mere message conduit.

Drawing parallels with advances in human-machine spoken dialog systems, we can see that incorporating intelligence into a spoken language based communication mediation system requires, among other things, careful user modeling in conjunction with an effective dialog management. In general, user modeling in systems design has been attempted at different levels and using a variety of approaches. Rich (3) has proposed a 3-dimensional space to describe the relationship between user models, defined as the knowledge about people, and their uses. In Table 4.1 the three axes of these descriptors relate to the size of the population the model describes, the fashion in which the model is created and also the temporal scale the model is attempting to characterize.

While there has been a fair amount of excellent user modeling work in the context of human-machine spoken dialogs including user simulation (55; 56), reasoning about a user's goal or intention (57), user expertise modeling (58), and evaluation techniques (59), relatively little effort has been devoted in this regard on machine mediated human-human cross-lingual dialogs, the topic of this chapter. The motivation stems from the need for informing designs of speech translation

systems for their increased effectiveness and usability as communication aids.

Construction of a user model based on the desired user features, however, can be a daunting task. Generally, two approaches – "Profiling modeling" and "Statistical modeling" - are widely used in this regards. The profile acquired from a user can be used for generating an appropriate system response, such as personalized search (60), or in providing appropriate help to the user when needed (57; 61; 62). In this present work, we adopt the second approach, where predictive statistical user models are derived from usage data. It is considered a powerful approach to model user behavior (63) and its effectiveness has been demonstrated by previous research (58; 64). We specifically propose a Bayesian network user model for our analysis to exploit its effective reasoning capabilities under uncertain situations.

In order to study user modeling issues in speech-to-speech translation systems, we consider two separate but mutually dependent channels – the Human-Machine-Human (machine mediated) and the direct Human-to-Human (interpersonal) channels. The verbal communication is handled through the machine, and effects of uncertainty and errors in the machine processing can be expected to be predominantly manifested in the verbal behavior of the user. On the other hand, the interpersonal channel is characterized by direct gestural non-verbal exchanges (such as head nods) as well as indirect verbal means (such as through adaptation to one others speaking styles). Our analysis in this chapter is restricted to aspects of the verbal behavior in these channels. The rest of the chapter is organized as follows. After a description of the speech-to-speech system used in this study for doctor-patient interactions and the corresponding data in Section 4.1, in Section 4.2 we analyze and model user behavior in the mediated channel under potential uncertainty by focusing on the "Retry"(*Repeat/Rephrase*) behavior. We describe a dynamic Bayesian model to predict such behavior and evaluate its performance in offline data. In Section 4.3, we present an online experiment with agent feedback and report the results. Finally, conclusions and a description of future work plans are given in Section 4.4.

4.1 System and Dataset

4.1.1 A Two-way Speech Translation System with a push-to-talk interface

The system used for the study of this chapter is a Speech-to-Speech translation device that facilitates two way spoken interactions between an English speaking doctor and a Persian (Farsi) speaking patient (51). This version of the system uses a push-to-talk modality to initiate a speaking turn which has its advantages and limitations. The push-to-talk interface minimizes recognition and translation errors since users can verify concepts before executing the final decision for "speaking out" the translation but has the disadvantage of creating less spontaneous and less natural interactions.

Furthermore, the goal of the system is to facilitate a task oriented rather than a free-form social interaction between the two participants. Specifically, the domain of usage of the system under study is task-specific (or goal-oriented) interaction between a doctor and a patient. It is within this context, the system design strives to achieve not only optimal technology performance, such as of automatic speech recognition and translation, but also maximal user satisfaction. Prior work has clearly shown that user satisfaction is one of the most important efficacy metrics of medical domain interactions (65; 66).

A functional block diagram of the system used in the present study and its data flow are shown in Figure 4.1. The user's spoken utterance is converted into textual form by an automatic speech recognizer (ASR) in the appropriate language of the speaker (English for the doctor and Farsi for

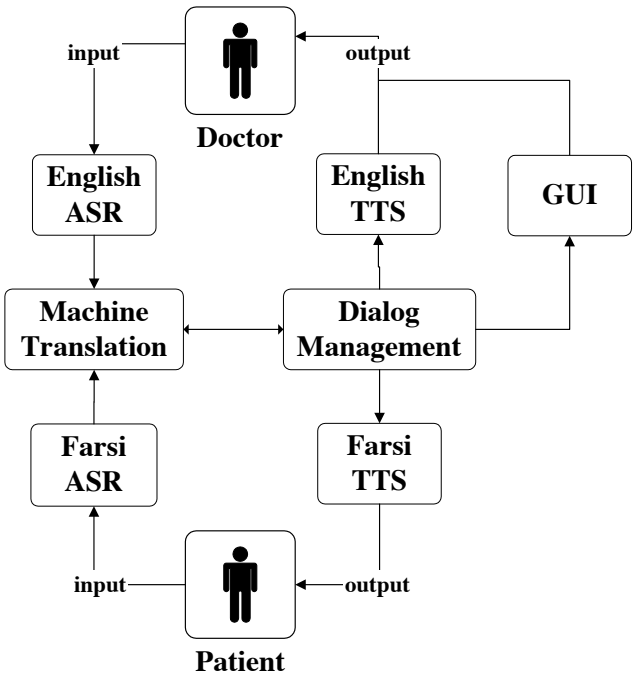


Figure 4.1: Simplified data flow diagram of our two way speech translation system for doctor-patient interactions. English and Farsi Automatic Speech Recognition(ASR) models get the input from users (doctor and patient, respectively) while the Machine Translation(MT) module is responsible for automatic translation and classification of the input. The Dialog Manager(DM) manages the interaction and communicates the translated results to a graphical user interface (GUI) and a text to speech (TTS) synthesizer (in English and Farsi as appropriate).

the patient in this case) and further processed by two parallel mechanisms: one by a phrase-based statistical Machine Translation (MT) module that translates the text from one language to another

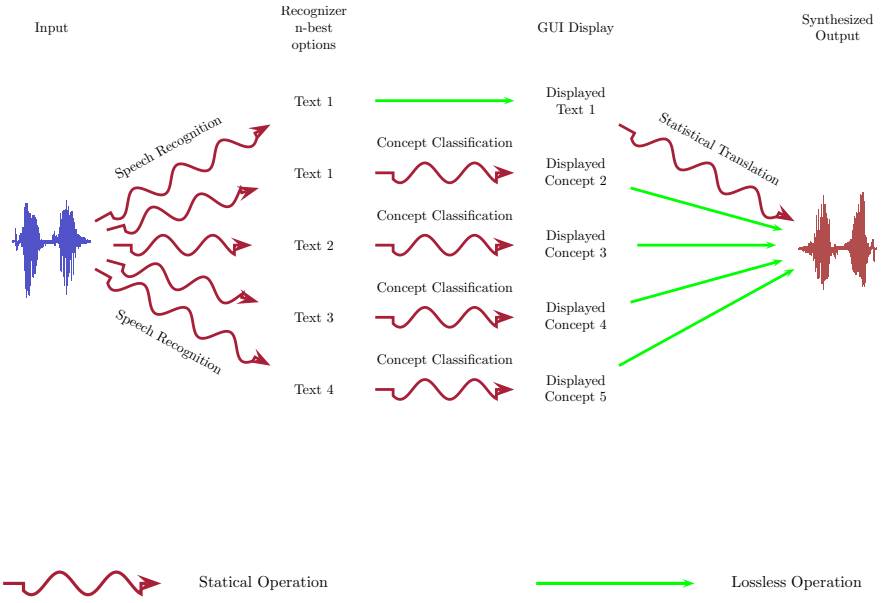


Figure 4.2: The internal procedure of generating speech translation hypotheses in our system. Two parallel mechanisms are implemented. In the first one, the topmost recognition candidate i.e., the first-best choice of the ASR – that has already gone through a lossy speech to text mapping process – will go through another lossy operation – the statistical translation. In the second one, that utilizes an utterance classifier, the top four recognized candidates from the ASR (the so called four-best results) are mapped into conceptual classes, also a lossy operation, but the canonical form result – after both lossy operations – is the one displayed on the screen for the doctor's choosing.

and the other by a statistical classifier which attempts to categorize the utterance into one of several predetermined "concept" categories. The Dialog Management (DM) module interacts with the MT/classifier and the GUI and TTS modules to deliver the data to the user. In the system of this study, the visual output provided by the GUI is made available only to the (English-speaking) doctor, who is assumed to have the primary control of the interaction.

To better understand the translation device operation, and the associated issues, we can identify three distinct operations in the process that can introduce uncertainty into the communication chain. The first inherently lossy operation is the conversion from speech into a textual transcription of the spoken utterance through statistical pattern recognition (ASR) i.e., often the transcript may not accurately represent what the user spoke, characterized by deletion/insertion/substitution of words. The second one is the translation. We have two concurrent statistical approaches to this step (statistical machine translation and an utterance concept classifier) that represent a lossy mapping. The third stage is the conversion of the target language transcript from text to audio by synthesizing the speech, through Text-To-Speech (TTS) synthesis which can be lossy due to several reasons including due to operating on the noisy output from the ASR and translators. All these potential information losses can impact the communication between the participants.

By design, the interface control of our experimental system was asymmetric in the sense that the (English-speaking) doctor had exclusive control over the interface, and access to the GUI, while the (Farsi-speaking) patient did not. This was to allow even untrained and non-educated patients access to the system. The system allows for the doctor to decide whether to transmit one of the

several alternate hypotheses offered by the system to the patient or reject all of them (repeat or rephrase). Some of the options provided to the doctor can be seen in Figure 4.3 and the hypotheses belong to one of two classes:

1. The first is the English transcription of what the machine thinks the user said. The machine does not provide a translation on the screen (presumably it would not be useful for the doctor who doesn't know Persian) but a statistical phrase based translation would be provided to the patient if the doctor chooses this option. However, such statistical machine translation *can not* guarantee accurate translation of the displayed text. This option mainly allows the user to detect errors from the ASR stage of the translation process, and thereby reducing the risk of error during the translation.

2. The second category of options takes the recognized transcript (output of ASR stage) and maps it into one of over several pre-determined concept categories. These categories were manually specified and for this domain there were about 1200 concepts. This mapping operation from text to concept is also lossy, but unlike the first hypothesis, since these concept categories are pre-programmed in the system, a back-translation (canonical form) in the language the doctor understands can be displayed for the doctor's choosing. This means that what the doctor sees on the screen already includes any errors likely made by both the ASR and translation steps, and that the translation the patient will hear will be lexically identical to the hypothesis displayed on the screen. Figure 4.2 depicts these procedures conceptually. It is clear that if one of the canonical sentences is satisfactory from a concept transfer perspective, it should be the best choice for the user since these guarantee accurate translation.

Users of the device were encouraged to employ the second category of options (labeled on the GUI: "I can definitely translate these") if these options were deemed valid representations of their utterances, rather than the first option (labeled on the GUI: "I can try to translate this"). For example, in Figure 4.3 when the doctor says "You have fever?" the device can try to translate the ASR text output "You have fever" or it can definitely say "Do you have a fever?", the surface form for a concept category related to "fever-inquiry".

The monolingual patients on the other hand are assumed to be untrained in using the system – and to ensure uniform results in the experiments described in this chapter – are not allowed to see the screen. The system decides, based on confidence scores of automatic utterance to concept classification, whether their utterance is close enough to a particular concept class. If deemed confident, the cluster-normalized form concept will be transferred to the doctor, and if not a direct potentially noisy statistical translation of the text will be provided. Most of the time an incorrect transfer can be detected by the doctor due to the lack of coherence with the discourse of the interaction. The Persian patient can also choose to request, verbally or through gestures, repetitions or repairs if they so chose. Note that an experienced doctor, in the case of receiving information that does not match the discourse can assume that he needs to do error control by rejecting the solution provided by the system (and repeat/rephrase).

In terms of component level performance of the system used in the present study, the ASR word error rate, the concept transfer rate and the IBM BLEU translation score are given in Table 4.2. These results stem from the evaluation done under the DARPA Babylon program. The overall concept transfer rate of the system is 78% – this denotes how many of the key concepts (such as symptom descriptions) were correctly transferred overall in both languages according to human observers for the 15 sessions examined in this chapter. Also, in the Table 4.2 the word error

Figure 4.3: Transonics system screen GUI. After speaking, the user(doctor) can choose one of several hypotheses presented on the GUI.

Table 4.2: DARPA evaluation on medical domain for the speech translation system of this chapter. Component and Concept measures as: ASR word error rate (lower is better), SMT BLEU score (higher is better) with the clean text transcript input or with the ASR output as an input.

DARPA Evaluation results		
	English	Persian
ASR WER	11.5%	13.4%
	English to Persian	Persian to English
IBM BLEU (text)	0.31	0.29
IBM BLEU (ASR)	0.27	0.24
Overall concept transfer		78%

rate(WER[1]) and the IBM BLEU[2] scores are provided.

4.1.2 Data-set

The data analyzed for the user modeling purpose are from 15 interactions between doctors and standardized patient actors. Both the doctors and patients are monolingual and, in addition, acoustic masking was in place to ensure translations are only being transferred through the device. The spoken interactions were logged by the system and also transcribed manually. Automatic logs contain recognized utterances (hypotheses) of the ASR, all translated hypothesis from the translation component (both SMT and classified concepts). These come with the confidence levels and the system procedure information.

[1]Word Error Rate is the sum of the number of words in error (substitution, deletion and insertion) divided by the number of words in the reference transcription.

[2]In simple terms, the more ways a certain utterance can be translated, the lower will be the maximum possible score, since one translation will be compared with many possibilities. So although the score is on a theoretical scale of $0 \leq$ IBM BLEU ≤ 1, even the best human expert translators can only achieve average ranges of near a half of that.

Table 4.3: Table shows a simplified portion of the data log acquired automatically by running the Transonics speech translation system. There are system routing tags(FADT, FDMT, FMDT, FDGT, FDGC, FGDT – F: Flow, A: Audio server, D: Dialog management, M: Machine translation, G: Graphical User Interface, T: Text, and C: Control) indicating the data flow from/to on the left side and the data being processed on the right side. Actual data are in the content column. Additional information logged, not shown for simplicity, include time stamps, utterance sequence, confidence and class numbers.

System Routing Tag	Content
FADT	YOU HAVE OTHER MEDICAL PROBLEMS |
	DO YOU HAVE OTHER MEDICAL PROBLEMS
FDMT	YOU HAVE OTHER MEDICAL PROBLEMS
FMDT	SmA mSkl pzSky dygry dAryd |
	YOU HAVE OTHER MEDICAL PROBLEMS
FDGT	YOU HAVE OTHER MEDICAL PROBLEMS
FDMT	DO YOU HAVE OTHER MEDICAL PROBLEMS
FMDT	VyA hyC mSkl pzSky dAryd |
	DO YOU HAVE ANY MEDICAL PROBLEMS
FDGT	DO YOU HAVE ANY MEDICAL PROBLEMS
FDGC	ShownAllOptions
FGDT	Choice*1

Automatic tagging of the retry behavior was made possible through system logs, and the speech recognition WER scores were acquired by comparing automatically recognized utterances and their human generated transcriptions. It may be interesting to note some relevant information regarding the data characteristics. The average number of turns (each turn is a doctor or a patient utterance) in a conversational dialog is 30.13, with a slightly higher number (33.46) for the doctor than for the patient (26.8) with standard deviation of 8.7 and 10.6 respectively. The longest utterance was 13 words long for both the doctor and patient side, while on average utterance length was 4.45 and 2.42 words for the doctor and patient, respectively. The shorter average utterance length of the patient reflects the fact that a significantly large number of their answers were short, such as yes/no answers. The total time for the whole data set is 4 hours.

Because of the dynamics created by the push-to-talk interface (managed by only the doctor), the doctor-side data contains abundant information we can utilize to model user behavior in the mediated (verbal) channel.

4.2 The Mediated Channel

We refer to the information path between the two participants through the machine as the *Mediated Channel*. In this channel, a user is cognizant of the machine and acts by considering both the response of the system and his own prior actions. Also, the system can detect how a user behaves or what information is going through the channel. In this sense, it can be regarded as similar to a Human-Machine interaction scenario.

The methods of identifying the user's model from interactions with a device include investigating behavior patterns (67; 68) and stereotypes (69). Following these generally classified assumptions, considerable research efforts have been undertaken covering various topics and systems: Komatani (58) introduced a general user model with skill level, knowledge level, degree of urgency in a spoken dialog system, Carberry (70) modeled user preferences in a natural language consulta-

tion system, Conati (71) proposed how to manage uncertainty in a student model by performing assessment and recognizing plans for a tutoring system, and Prendinger (72) utilized physiological data for determining affective states for an emotion recognition system. Furthermore, some frameworks have been suggested for rapid and efficient implementation of user models such as in (73; 74; 75).

Error handling mechanism is an important aspect in the design and optimization of a spoken dialog system. As mentioned earlier the spoken communication channel between a human and a machine is inherently noisy, which can further be exacerbated by user-dependent uncertainty such as due to limited world or task knowledge. The significance of considering user behavior under problematic conditions in human-machine interaction is demonstrated for example by our prior work (76), where we highlighted the importance of repeating and rephrasing cues. Similarly, the work of Batliner (77) utilized the features such as prosody and linguistic behaviors to model and recognize trouble in communications. Detection and modeling of problematic communication conditions helps to prevent and recover from errors effectively.

Specific user behavior patterns can be attributed to specific user types. Similar to the notion of expert/novice users, in this work, we consider the idea of identifying accommodating and non-accommodating ("picky") user types under problematic interaction situations with the motivation that distinct interface strategies can be developed for each case. Our experimental analysis indicates that for the same average speech recognition WER, one user retried 95% of the time while another user only 65%. For example, we have observed that certain users are more accepting of minor errors in translation and recognition (e.g., function word insertion such as in "And do you have fever?" when they actually spoke "Do you have fever?") while others completely reject such a hypothesis from the machine as not their intended utterance, despite the fact that it conveys for all practical purposes the identical meaning.

We therefore propose modeling users in one of three categories(*Accommodating, Normal and Picky*) based on the analysis of the active participant, the doctor. Following which, we train a system that can detect in which category the user belongs based on the user behavior through the interaction history and current utterance features. While devising specific interventions based on the model outcome is not the goal of this chapter, we hope that this approach will however enable future research in building agents that can appropriately adapt the system according to detected user behaviors similarly to what previous studies have demonstrated (58; 78; 79).

4.2.1 Analysis of repeat/rephrase("Retry") behavior

Repeat or rephrase (Retry) was the primary user behavior observed under problematic conditions caused by non-optimal or poor system performance in the Transonics system. In addition to the user type being an important factor in determining the degree of retry actions, the level of speech recognition error was found be an important factor. However, in our *standardized subject*[3] experiments, the difference range of the speech recognition error among users is small, therefore we assume that the user type has a stronger effect on the observed retry behavior. In addition to the small variance in the speech recognition error, we observed that most of errors stem from insertions of function words and that keywords are mostly correctly recognized. Typical examples of errors with erroneously inserted words underlined are: "A how are you", or "tell me THE about your pain". Other potential contributing factors such as user's emotion, knowledge, gender, physical condition, hastiness, etc. are not considered at this stage, but are of interest and will be included in the analysis once larger data sets become available.

[3]The subjects were all native U.S. English speakers, medical professionals and trained equally before using the system.

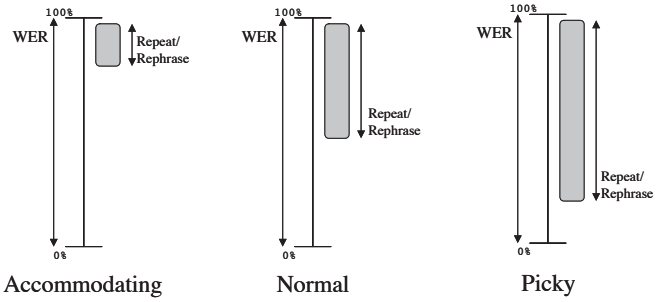


Figure 4.4: The *Accommodating* user tends to "Retry" significantly less than the other users while the *Picky* user tries significantly more. A user in between these extremes is defined to be a *Normal* user. WER is the speech recognition Word Error Rate and the above graph semantically demonstrates the ranges of WER for which each user type tends to "Retry."

4.2.1.1 Categorizing User types: Accommodating, Normal and Picky

User type is a casting of a user along several categories; it can be based on demographic information, such as *Gender* or *Age* or a heuristic category such as *Expertise* or *Knowledge level*. We consider, in this chapter, the degree of user's accommodation to spoken language processing errors as the criterion to decide a user type. The use of such heuristic domain-specific criteria has been prevalent in user modeling research. For instance, in (58), user skill level is defined by the maximum number of slots filled by utterances and in (71; 80), knowledge level is decided based on correct answers to the domain questions. In most cases, heuristic methods are used for user type classification even though those may not always be too accurate – for example, if we assume that knowledge level is judged by the number of correct answers to system questions, this is usually a good metric, but not a perfect one since the user may give wrong answers on purpose to trick the system, may be tired and not pay enough attention, or may not be motivated enough to devote the necessary attention.

For our off-line model, we cluster user types based on the total number of retries of each user. We assume that accepting different ranges in WER depends significantly on the user type, as conceptualized in Figure 4.4, and hence we define

- *Accommodating*: users tend to accept highly erroneous transcriptions compared to other users.
- *Normal*: users accept some degree of errors
- *Picky*: users tend to reject all but the most exact transcriptions, thus being very strict in what they accepted for translation.

Based on data from the 15 sessions analyzed in this work, we clustered the users with the k-means algorithm into the 3 classes as shown in Figure 4.5. Note that one could argue in favor of fewer or more quantization steps along the accommodation axis. Such decisions depend more on the action to be taken upon classification, and the available data for the analysis.

From the clustering results, 7 (47%) users present themselves as accommodating, 5 (33%) as normal and 3 (20%) as picky. The users tend to *retry* at different degrees: *Accommodating* 19.3%, *Normal* 31.3%, and *Picky*: 40.7%. The average WER rate across *all* the utterances, however, does not vary significantly and stands at 35.9, 43.8 and 38.7 for *Accommodating*, *Normal* and *Picky*,

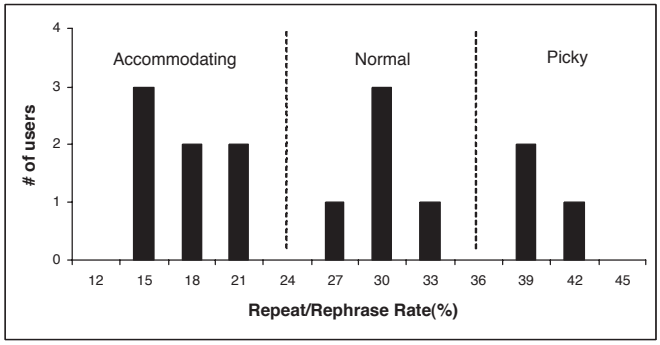


Figure 4.5: The quantized retry rate over 15 interaction sessions on the doctor side. The criteria (average retry rate) based on the data analysis led us to categorize the users into 3 types: Accommodating, normal, and picky.

respectively. Hence we did not employ WER as a feature for the clustering of user types. Note that although the average WER is relatively constant from user to user, the error that users consider acceptable is not, as demonstrated by the variable degree of retries.

Assuming a certain threshold separating the High-Quality (HQ) speech recognition performance from a Low-Quality (LQ) performance (a detailed discussion of how the two regions of performance can be decided is provided in the next section, Sec 4.2.1.3), we empirically acquired the Conditional Probability Table(CPT) over all the 15 interactions as shown in Figure 4.6. We can clearly see the difference in user accommodation when operating in the LQ region.

When the condition represents relatively high system performance (HQ performance), other behaviors ("Accept") dominate covering over 90% in most cases, and allowing us very small amounts of data for observing the "Retry" behavior.

4.2.1.2 User behavior model with the Transonics system

Since in our analysis we observed that the system error alone can not account for the large variability in user actions, we hypothesize that the user type combined with the system error under problematic conditions affects the retry behavior. The following conditions are assumed: 1) The system is stationary and the performance is shown in the Table 4.2; 2) The subjects are native speakers(U.S. English) and user performance is consistent in terms of machine recognition (no acoustic/lexical mismatch issues in speech recognition); 3) Domain knowledge of subjects is the same (all medical professionals) 4) Skill and adaptation levels are expected to be the same based on the given environment (trained with equal time and materials and provided the same experimental environment for equal time).

4.2.1.3 Threshold of high/low quality system performance

Another important issue we need to deal with is the threshold of average acceptable WER for each user. This is a complex issue that is related to each user's personal preferences and traits. We empirically approached this problem with the relative WER average based on retry and accept behaviors across all other users. We assume that a user retries if the system performance falls

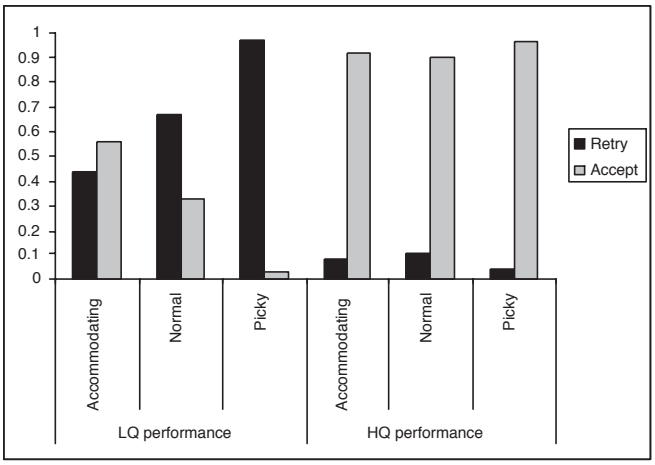


Figure 4.6: Conditional Probability Table(CPT) over user behaviors(discrete) – "Retry" and "Accept". Each user type is represented numerically with regard to Low Quality(LQ) and High Quality(HQ) system performance(recognition error rate). The Y-axis represents the probability of user behavior conditioned on user type and system performance.

below a threshold, thus we clustered the per-utterance WER into two groups: the group of accepted utterances and the group of the utterances that are rejected. The Low Quality(LQ)/High Quality(HQ) performance threshold is the separating point of the two clusters, at a WER of 56% for the data of these 15 interactions. This implies that there is a high probability of a retry if the WER increases above 56%. For training and testing purposes, the threshold is acquired in a n-fold cross-validation from 14 interactions and tested on the remaining 1 interaction. Note that although the threshold WER may seem to imply a very low accuracy for allowing a concept transfer, the classifier frequently may allow accurate concept transfer with WER much higher than that if a keyword has been recognized correctly and the classification gave at least one option which is valid. For example: "Are you having a headache now?" will have a classifier top choice of "Do you have a headache?" even if only the word "headache" has been correctly recognized by the ASR.

4.2.2 A dynamic Bayesian network user behavior model

A dynamic Bayesian network is a promising representation for modeling the inter-causal relationships of "Retry" behavior with temporal information. The promise of this model has been highlighted in the user modeling field across various applications. The Lumiere project (57) utilized Bayesian models for capturing the uncertain relationships between the goals and needs of a user. Conati (71) used Bayesian network to model a student for an automated tutoring system which assesses the knowledge, recognizes plans and predicts actions of each student. Recently, Grawemeyer (81) modeled users' information display preferences by using Bayesian reasoning. Also, the theoretical benefits in its performance and extensibility as a classifier have been thoroughly described in (82).

In spite of their remarkable power and potential to address inferential processes, there are some inherent limitations and liabilities to Bayesian networks. First, a Bayesian network cannot represent every possible situation (uncertainties and dependencies) and it takes a long time to

Table 4.4: User type inference algorithm computes the probability of user types, *Accommodating*, *Normal* and *Picky* respectively. Each user type is predicted by Bayesian reasoning and updated until one of them becomes believable.

Input: User behavior("Retry" or "Accept") and HQ/LQ recognition information.
Output: The most believable user type
Initial: User types with the same probability
Step1: The probability of each user type is given by the Bayesian reasoning.
Step2: Update the prior of each user type
Step3: Check whether the belief of the highest user type probability is enough
Step4: If it is not enough to be believed, go to the Step1
Return A user type with the highest probability

choose necessary nodes for the network. Second, the prior knowledge (probability) of each node of the network may be biased depending on the measurement approach and this may distort the network and can generate unreliable response to a user. For example, in (57), experts constructed Bayesian models for several applications, tasks and sub-tasks by doing user studies however, that assumes sufficient and representative coverage of user activities in the observed data.

The details of the proposed DBN implementation are presented in the following sections and general user type prediction algorithm is given in the Table 4.4.

In this analysis the variables of user behavior (retry/accept) and the system feature, the utterance confidence score (or for off-line processing WER), are the observed variables and the user type, the unknown variable. In the design phase, the network is built by learning parameter values and interrelations of user type and observed variables.

The user type is assumed to be constant, despite the fact that some user characteristics may vary during the course of an interaction. For example, talkative people may be more reserved in communicating when depressed, tired or under stress. A person who is in general sensitive to any kind of system errors can ignore those when he/she is busy. In addition, we often observe that users take time to exhibit their steady state behavior due to an initial adaptation to the other entity, be that a human or a system. It is assumed that the executed behavior and observed feature value are the best representatives for the user type at each time and the model with these variables is extended dynamically with the temporal information.

We are operating under the assumption that information about the user type could help in altering the system strategy. In addition, this strategy enhances the experience of the user-machine interaction similar to the use of expertise model developed in previous efforts and employed in efficient system strategy design (58; 79).

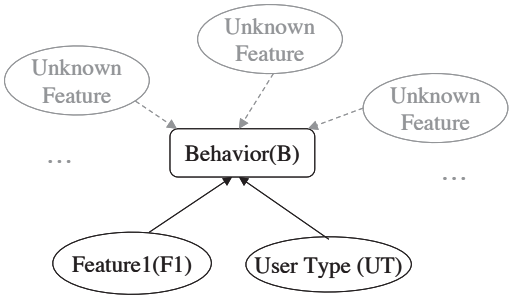


Figure 4.7: A generic directed graphical model; the Bayesian network represents the relation in which a user behavior(B) is influenced by a user type(UT) and a feature (F1). There may be unknown features such as emotions and skill level but only one feature is considered for the suggested model.

4.2.2.1 A model of user behavior over a single iteration

We quantize the variables of user type (UT), behavior (B), and system accuracy (F)) and these satisfy:

$$\begin{aligned} \sum_{i=1}^{n} P(UT = ut_i) &= 1 \\ \sum_{i=1}^{m} P(B = b_i) &= 1 \\ \sum_{i=1}^{k} P(F = f_i) &= 1 \end{aligned} \tag{4.1}$$

where we chose $n = 3$ discrete levels for the user type, $m = 2$ for behavior and $k = 2$ for the WER. Note that we represent variables by an upper-case letters (e.g., UT, B, F) and its values by that same letter in lower case(e.g., ut, b, f).

The Bayesian network in Figure 4.7 shows the complete directed graphical model (static) with the relations among a specific behavior, user type, and features (including unknown features).

Multiple features can exist and each can have different effect on the user behavior. Prior work has demonstrated that fewer features are better for improved accuracy/performance (83), particularly in small data-sets. Also, unimportant features can be eliminated by utilizing probabilistic measures related to the features (84). In the design of the suggested Bayesian model, we chose to incorporate only one feature due to the small amount of data: the quantized (HQ/LQ) WER variable is incorporated with an independent user type variable.

Based on this general procedure, an actual sequence of stepwise conditional probabilities is formed as in the equation (4.2) with the random variables of parents (UT and F) and a child(B). In the user behavior model, we assume that there is no relationship between user type and feature.

$$P(B, UT, F) = P(B|UT)P(UT)P(B|F)P(F)/P(B) \tag{4.2}$$

where, B = user behavior, UT = user type, F = feature.

Once the network structure is defined and the conditional probability is decomposed, the quantization of the data in the chosen levels needs to take place. In the suggested model, we have 2

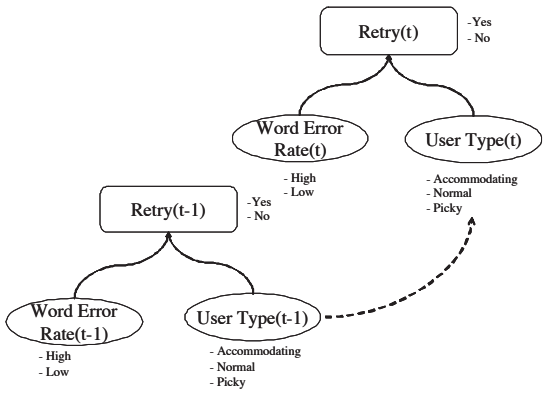


Figure 4.8: A dynamic Bayesian network is used to infer a user type over time in the mediated channel. The belief of a user type becomes strengthened as the interaction progresses.

discrete levels for user behavior (retry/accept) and system performance (HQ/LQ) and three user types (*Accommodating*, *Normal* and *Picky*). To give a value for each discrete level, we can utilize a domain expert's knowledge or learn it from the data-set. The second method is adopted in this experiment and the values are learned in a n-fold cross-validation from the training data-set (using 14 out of 15 interactions) for testing on 1 interaction allowing for presenting averaged results over a total of 15 experiments for the 15 interactions in the corpus.

4.2.2.2 A dynamic model – temporal belief reinforcement

In reality, it takes time to grasp an accurate user type by observing user behaviors and factors (features). For example, by observing a one-time accommodating behavior of a user is not enough to decide a definite user type while the observation of some consistent behavior over time strengthens the belief of the user's type. This idea is formulated as a dynamic Bayesian network (DBN) shown in Figure 4.8. The user type transition mechanism from time $t-1$ to t is supported by the Markovian property that the conditional probability of the current user type(t) depends on the previous user type($t-1$) and it includes the history implicitly by this assumption.

During training, we employ the complete interaction to reason on the user type by using the Maximum Likelihood Estimate (MLE) as in equation (4.3).

$$P(B|F,UT) \quad = \quad \frac{P(F,UT,B)}{P(F,UT)} \tag{4.3}$$

where, $UT = \{ut_1 \dots ut_n\}$, $B = \{b_1 \dots b_m\}$, $F = \{f_1 \dots f_k\}$.

The prior for the feature, Word Error Rate(WER) is also acquired from the training data and the prior of the user type is initially set equally distributed and updated dynamically.

In the absence of large amounts of training data, unconstrained identification of the priors of transition probabilities in a data-driven fashion is not feasible. We instead place parametric constraints on the transition probabilities and identify these parameters in a data-driven fashion. The parameters are the probability of:

- Staying in the same type. This probability is expected to be the highest. (P_{SameType})

Table 4.5: Values of transition priors. The parametrization allows 4 variables to represent nine time-varying priors, thus allowing estimation from limited data.

	UT^t_{Acc}	UT^t_{Nor}	UT^t_{Pic}
UT^{t-1}_{Acc}	0.90	0.05	0.05
UT^{t-1}_{Nor}	0.05	0.90	0.05
UT^{t-1}_{Pic}	0.05	0.05	0.90
λ		0.05	

- Transitioning across adjacent types (Normal to/from Accommodating and Picky). ($P_{\text{WithNormal}}$)
- Transitioning across opposite types (Accommodating to/from Picky). Expected to be the lowest probability (P_{Opposite})

In addition we define a parameter that reinforces beliefs over time by modifying each of the above probabilities and is defined in terms of the ratio:

$$\mu = \lambda \frac{(\text{Turn Number})}{(TotalNumberofTurns)} \tag{4.4}$$

where λ is expected to be a very small number because we want smooth increase of the same user type transition probabilities over time. Resulting in:

$$\begin{aligned} P_{\text{SameType}}(n) &= P_{\text{SameType}}(n-1) \times (1+\mu) \\ P_{\text{WithNormal}}(n) &= P_{\text{WithNormal}}(n-1) \times (1-\frac{1}{3}\mu) \\ P_{\text{Opposite}}(n) &= P_{\text{Opposite}}(n-1) \times (1-\frac{2}{3}\mu) \end{aligned} \tag{4.5}$$

Note that the probabilities are normalized in each turn.

Table 4.5 presents the values of the parameters. We can also observe that over time the probability of transitioning across opposite types will decay faster than the probability of transitioning across adjacent types.

To infer a user type, the posterior probability of user type conditioned on behavior and feature is computed as in Equation (4.6) by applying Bayes' rule.

$$P(UT|B,F) = \eta P(B|UT,F)P(UT) \tag{4.6}$$

The user type is independent of the observed feature therefore $P(UT) = P(UT|F)$, while $\eta = P(B|F)$ plays the role of a normalizing factor, ensuring that probabilities of user types sum to one.

At each turn, by maximizing the probability of each user type(ut_i) as in Equation (4.7), we obtain an estimate of the most probable user type, however the decision is not made until confidence in the belief of user type is significant.

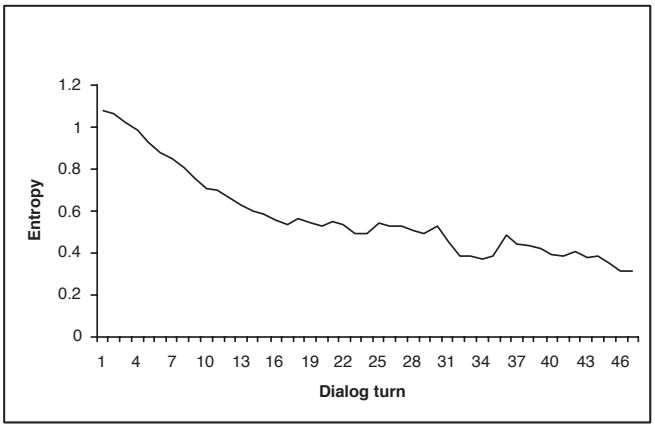


Figure 4.9: Entropy of three user types becomes lower as the dialog turn increases. The threshold of deciding the final user type can be set based on this tendency under a dynamic Bayesian reasoning.

$$\text{argmax}_i \; P(ut_i|B = b_1, F = f_1) = \text{argmax}_i \; P(B = b_1|ut_i, F = f_1)P(ut_i) \quad (4.7)$$

where, b_1 = an evidence of the user behavior, f_1 = an evidence of the feature.

In identifying when a decision on the user's type can be made, we need to consider an acceptable confidence *"Threshold"*. This includes two dimensional conditions, when and how to draw a conclusion from the inference. One approach is to decide the final user type when all the available data has been processed (the last state of the DBN) and the evaluation in section 4.2.3 is based on this method. An alternative approach is maximum entropy, a good measure that has been utilized in previous work to classify user behaviors (68). This may be a more objective and concrete measure of convergence and more appropriate for real-time implementations. As in the Figure 4.9, we can see the tendency of decreasing entropy for the user type probabilities over all 15 interactions. The entropy decreases as the DBN converges and a lower entropy means that the intra-speaker probabilities of user type are more discriminating. To utilize this mechanism, we could set a certain threshold below which a decision would be made. Otherwise, a user type would be labeled as still unpredictable or not inferable.

4.2.3 Model Validation

We evaluated the automatic identification of the user type by employing the n-fold validation, thus using 14 interactions for training and one for testing, and performing a total of 15 experiments. The goal was to identify user type through the interaction data. Priors were set to be equal (0.33) for the three user types. The classification was successful in 13 out of the 15 dialogs examined by assuming a convergence of the DBN at the end of the available data (method 1, described above). Both errors occurred in identifying the normal user type, and in both cases it was clear that convergence had not been reached. The DBN was fluctuating between *Normal* and *Picky* in one case and *Normal* and *Accommodating* in the other case. We believe, that this may reflect a

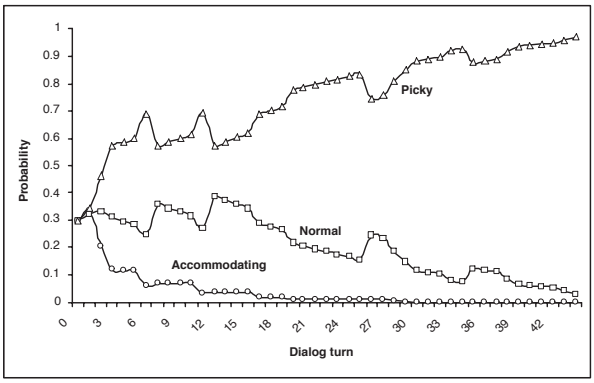


Figure 4.10: The belief that the user type is "*Picky*" is strengthened over time in this example data set.

switching user behavior where, users may behave as picky (if the error is for example in a keyword) or as accommodating (if all the errors are in function words), or it may reflect users who exhibit behavior very close to the user type quantization boundaries.

In the following sections, two representative results of *Picky* and *Normal* user type inference by the suggested DBN model are presented.

4.2.3.1 Analysis of the *Picky* user type inference result

Dynamic inference results on an interaction (labeled as *Picky* type) that lasted over 44 turns is depicted in Figure 4.10. We can observe that the belief of the *Picky* user type is strengthened over time and is detected early on in the interaction. This implies that a user strongly follows a pattern, *Retrying* on most device errors and *Accepting* less when the system operates with high quality.

By observing the data of this interaction we can also note that this user (Figure 4.10) suspended the flow of conversation in many more cases compared to other users by being very selective.

4.2.3.2 Analysis of the *Normal* user type inference result

Figure 4.11 shows one of the most challenging users to classify in our corpus. The system in this case takes over 24 turns to eliminate the accommodating type, although it eliminated the Picky type from the 12th turn. Manual analysis of the data revealed that this user, despite being *Normal* in his average behavior, often exhibits *Accommodating* and sometimes *Picky* behaviors – crossing the boundary of two types, thus causing the DBN to take longer to converge.

4.2.3.3 Analysis of successful user type inferences

In this subsection, we present the analysis of successful user type classifications suggested by the model (13 out of 15 interactions in our dataset were successful). Figure 4.12 and Figure 4.13(b) represent the identification of the accommodating and picky user types. The correct user type is determined early in most cases (less than 10 interaction turns) even though some "Accommodating"

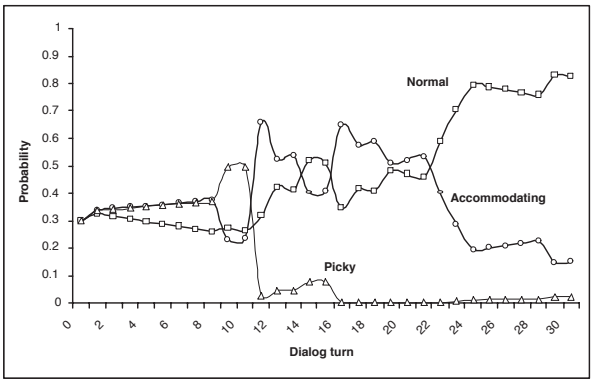


Figure 4.11: The belief that the user type is " *Normal*" is strengthened slowly over time.

users show different user types shortly in the middle of the whole interactions. The results imply that users in these two extreme types behave in their own style, especially, when the system performance is low. And, we can classify these two types early on by observing user behaviors and the system performance.

Different from the previous two extreme user types, the belief of "Normal" user type is gradually strengthened over turns by tailing off those of the other user types(Figure 4.13(a)). This implies that it took comparatively more time to be in middle point, in terms of the number of retry/accept under low/high system performance, between the two extremes.

4.3 Online evaluation of user model

In the following sections, we report the results of online evaluation of the user model using agent feedback. For this purpose, our new speech-to-speech communication system (called *SpeechLinks*) was used, and the English speakers' user behaviors were analyzed. The design considered the following: Picky users tend to reject even small recognition errors which do not affect the overall meaning transfer from user-spoken utterance in the source language to machine-generated utterance in the target language. In the opposite situation, accommodating type users tend to accept even critical recognition errors, which breaks natural conversations between users by causing incorrect meaning transfers through the device.

By providing agent feedback to users according to the user types, we could acquire better interaction efficiency (which will be defined in the result section) by encouraging users to change their behaviors in better direction.

4.3.1 Experimental setup

4.3.1.1 Participants and experimental domain

We recruited eight native speakers of English, four males and four females of ages between 20 and 28. All of them were undergraduate and graduate students at University of Southern California (USC). We also employed two Farsi speakers with some familiarity with the *SpeechLinks* project. Farsi speakers were one male and one female with the age of 21 and 24, and also undergraduate students. The choice of only two Farsi speakers familiar with SpeechLinks was made to reduce the variability space of the experiment.

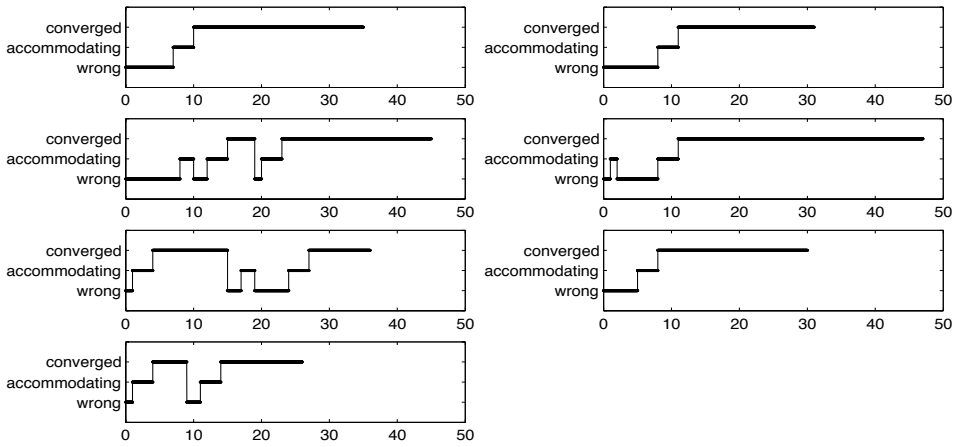


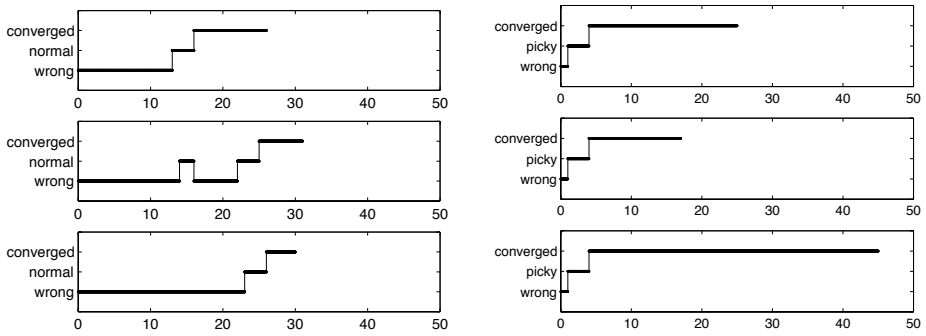
Figure 4.12: Inference on the data of various "Accommodating" user types in the corpus. X-axis indicates the dialog interaction turn. Y-axis indicates three levels of prediction results – wrong, accommodating, and converged to accommodating user types.

In total, 32 interaction sessions were collected from 8 native Speakers of English interacted with 2 native speakers of Farsi. For each interaction session, one native speaker of English and one native speaker of Farsi performed a diagnosis based on the provided scenario. The experimental time of each interaction session was approximately 30 minutes.

The domain of the experiment was medical diagnosis: Native speakers of English played a role of doctor and native Farsi speakers played a role of patient. Before the actual experiment, we gave one hour training session to English speakers and it included how to do a diagnosis of the disease with the supplied materials: the doctor's diagnosis manual table (a simplified example is shown in Figure 4.14 on the left) and the instruction of the experiment. The Farsi speakers were trained to use the system and to play the role of patient with the disease symptom card (simplified example in Figure 4.14 on the right). The purpose of this experiment was to study the English speaker behaviors reacting to agent feedback (driven by the proposed model) than to study Farsi speaker behaviors. The goal of the English speakers (in the doctor's role) was to find out a disease of a patient in each interaction session (The disease varies in each interaction session). Four diseases (flu, SARS, depression and hypertension) were used equally for the 8 English speakers during the experiment.

4.3.1.2 Scenario

The four scenarios were used in the same order during the experiment by each team (English-Farsi speaker pair). For each scenario, we provided a doctor's diagnosis manual table consisting of twelve (12) diseases in the column and related symptoms in the rows. The diseases in the column were: common cold, flu, food poisoning, lactose intolerance, depression, insomnia, hypertension, high cholesterol, liver cancer, lung cancer, SARS, and diabetes. The symptoms in the rows were, for example: 'chills' and 'fatigue,' and the number of the symptoms was 30, in which the actual symptoms were varied depending on the disease. We built this table as realistic as possible using



(a) Normal (b) Picky

Figure 4.13: Inference on the data of "Normal" and "Picky" user types over the dialog turns.

Diseases / Symptoms	Common cold	Flu	Food poisoning (Botulism)	Lactose intolerance	Depression	Insomnia
Abdominal pain	No	Mild, Upper left	Severe, Upper right	Severe, Upper left	Mild, Middle	No
Breathing	Normal	Difficult, Frequently	Difficult, Sometimes	Normal	Difficult, Sometimes	Normal
Chills	Slight, Frequent	Serious, Occasional	None	None	Slight, Occasional	Slight, Occasional
Concentration difficulty	Normal	Hard, Sometimes	Hard, Sometimes	Normal	Hard, Often	Hard, Often
Cough	Mild, Dry	Severe, Wet	No	No	Mild Dry	No
Diarrhea	No	Moderate, Sometimes	Intense, Frequent	Intense, Frequent	Moderate, Frequent	No
Dizziness	No	No	Severe, Irregular	No	Severe, Regular	Mild, Irregular
Exhaustion	No	Yes	No	No	Yes	Yes
Fatigue	Occasional	Often, Above the average	No	No	Often, Excessive	Occasional, Above the average

علایم عمومی
- تب بالا و سردرد

علایم مشخص (در صورت پرسیدن پزشک)
- خارش گلو
- احساس کوفتگی گاه به گاه
- خستگی خفیف، بعضی وقتها

علایم دیگر
- معمولی

Figure 4.14: Simplified example material: a part of doctor's diagnosis manual table for common cold (left). In the full size table, there are 12 diseases (column) and 30 symptoms (rows). A patient card for common cold is presented on the right.

the medical diagnosis information from **http://www.medicinenet.com.**

Farsi speakers (patients) were given a symptom card which provided only a few symptoms of the disease. On the right image in Figure 4.14, a symptom card for common cold is presented. We intentionally provided a few symptoms in each patient card to elicit more expressions from both speakers; English speakers needed to go through many combinations of diseases and symptoms in the look-up table to reason about a disease of the symptom card of a Farsi speaker.

Neither in the doctor role English speaker and the patient role Farsi speaker knew the disease name of each interaction session. We informed them of the disease names at the end of all four interaction sessions.

4.3.1.3 Experimental Procedure

The experiment was designed with two tasks, borrowing the idea of the evaluation method in the user modeling work by (58). Figure 4.15 shows this experimental procedure. In "Task A", native speakers of English performed the interaction session "without feedback" first and the session "with feedback" later. In "Task B", native speakers of English performed the interaction sessions in the

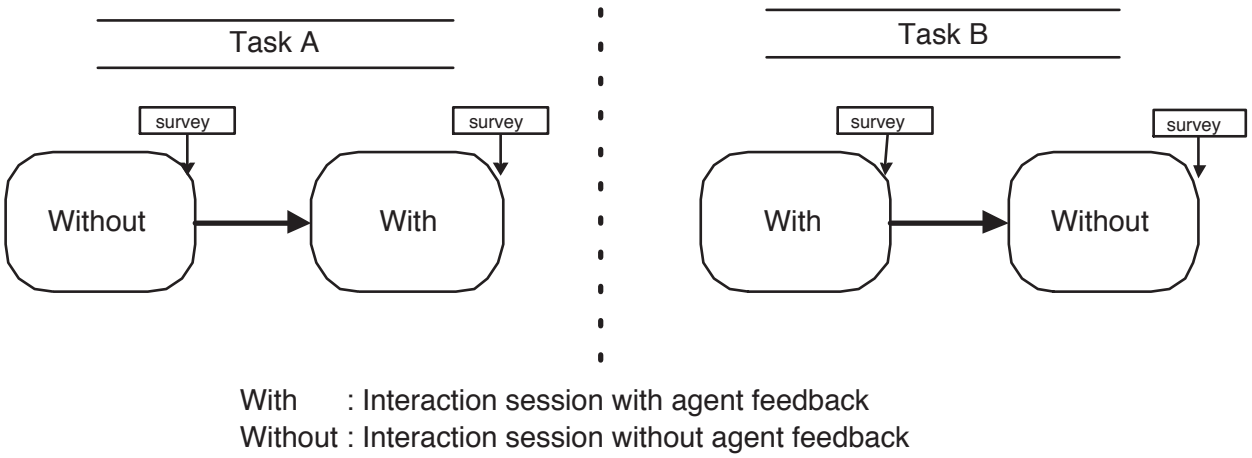


Figure 4.15: All 8 English speakers performed both "Task A" and "Task B" with 2 Farsi speakers in different ways: four of English speakers performed "Task A" first and "Task B" later, and the other four performed in the reverse direction. Each English speaker met different Farsi speaker in the different Task.

reverse direction. In each task, the English speakers interacted with different Farsi speakers – one male speaker for one task and the other female speaker for the other task. For the tasks, each English speaker visited the experimental room twice (two days). We assigned the Farsi speakers evenly to the two tasks: each Farsi speaker participated in "Task A" 4 times, and the "Task B" 4 times. In total, we collected 32 interaction sessions from this experiment.

For evaluation purpose, we collected five different survey questionnaires from each participant during the experiment. One is the initial survey about demographic information of the participant and user perception on many topics, such as user type and error tolerance level and past speech interface experience. After each interaction session, a questionnaire was given to each participant for the evaluation of system performance along multiple dimensions, such as user satisfaction and interaction efficiency. In total, 4 evaluation questionnaires were collected from each participant. Detailed analysis of questionnaires are provided in the section 4.3.2.3.

Each session lasted for thirty minutes approximately - we gave a 5 minute warning when the session was still continuing after thirty minutes. After finishing two sessions (with feedback and without feedback), participants gave us their opinions about the experiment.

All the interaction sessions were video taped. We analyzed the thirty two (32) interaction sessions in the video data in terms of identifying user types with their behaviors and, user behavior changes and system performance.

4.3.1.4 Agent feedback for accommodating and picky user types

Two different wordings of agent feedback were prepared for the two user types - accommodating and picky. When the system detected one of the two user types with high probability, it triggered the corresponding wording of agent feedback as in Table 4.6. The threshold of triggering an agent feedback was set as 0.65 which was acquired systematically from user training sessions. When the system detects either an accommodating or picky user type the first time, the wording (1) was presented to the users. After consecutive same user type identifications (e.g., three times), the

Table 4.6: Actual wordings of agent feedback for two user types. Two different wordings were used alternately for the same user type in case of triggering the same agent feedback over and over.

	For Accommodating User Type	For Picky User Type
(1)	"Consider rejecting bad options and rephrasing."	"Accepting system errors, if those have little impact on meaning, may improve system performance."
(2)	"The system is not always right. Some errors can cause significant degradation in your communication. When presented with bad options consider rejecting them and retrying"	"The system often inserts some additional words in its recognition results. Consider accepting some errors if those affect little the concept of the recognized sentence."

system changed the wording, in this case, the wording (2) was presented to the users. The agent feedback was presented to users in this fashion throughout the whole interaction session.

User type identification was conducted by dynamic Bayesian reasoning as introduced in the section 4.2.2. At each turn in the interaction, previous user behavior and ASR confidence level of the previous turn were utilized for computing the posterior probabilities of three user types. These probabilities were updated dynamically as the interaction proceeded.

The underlying assumption of the online experiment was that the ASR confidence level can be used to measure the ASR performance, which was measured offline by Word Error Rate (WER) as introduced in the section 4.2.2.1. The correlation between ASR confidence level and WER was mentioned and studied in (56; 85). ASR confidence level was computed using features at multiple levels, such as weighted acoustic model and language model scores.

4.3.2 Experimental results

We present the results of our online experiment using subjective and objective measures from various sources: user interview, questionnaire, video analysis and log data analysis. Statistical analyses were performed with SPSS 15.0.

4.3.2.1 Subjective measure 1: user interview

The interview with participants gave us insightful information about user opinions about agent feedback and its relation to system performance. Participants told us that the agent feedback provided hints when the interactions went wrong and it helped for smoother conversation flows and information delivery. In particular, the participants commented that agent feedback helped in mitigating frustration caused by repetitive errors. One of the picky type users said:

> "Agent feedback expedites conversation since users will not be repeating themselves in attempts to find an EXACT replication of their phrase."

Table 4.7: The statistics collected from the Likert-scale questions of the initial survey given to the participants. We measured users' own perception about their ability of dealing with general technology and speech interface, utterance length, and error tolerance level.

Likert-scale questions	mean	std. dev.
Speech interface experience(0:none - 10: more than ten times)	5.94	4.23
Inclination for the general technology (0: never comfortable - 10: comfortable)	6.81	1.51
Error tolerance level in the interactions with computers (0: not at all - 10: completely)	4.88	1.96
Error tolerance level in the communications with humans (0: not at all - 10: completely)	6.25	2.74
Utterance length (0: terse - 10: lengthy)	5.88	1.82
Hasty level when using computers (0: not at all - 10: completely)	6.44	1.41
Ability to work with computers (0: worst - 10: best)	5.63	1.31
Today's feeling (0:bad - 10:good)	7.63	1.20

4.3.2.2 Subjective measure 2: video analysis

By analyzing the video data of 32 interaction sessions, we subjectively identified user types of 8 English participants: 7 participants were picky and 1 was accommodating. For this identification, we specifically investigated the behaviors of users when the machine-recognized utterances have functional words which do not affect on the whole meaning of the utterances.

The analysis of video data suggests a trend of user accommodation to system functionalities and errors. We observed that the participants became accustomed to agent feedback in the early turns of the interaction session, and in the later turns, they did not pay attention to the agent feedback. We conjecture that they already knew what the agent feedbacks were and when the agent feedbacks would be triggered. From this viewpoint, the users of "Task A" (interaction session from 'with feedback' to 'without feedback') seemed to cope with system errors better than the users of "Task B." More analysis in this regard is presented in the following section.

4.3.2.3 Subjective measure 3: questionnaire analysis

We collected five questionnaires from each participant and the Likert-scale questions were given to the participant. The initial questionnaire was intended to measure users' own perceptions about their ability to deal with general technology and speech interface, utterance length, and error tolerance level (Table 4.7).

One finding from the initial questionnaire is that some users did not have speech interface experience at all while others had some experience. To reduce this gap, we gave a one-hour training session to all participants, which included how to use the system. Another interesting finding was that the error tolerance level in the communication with human was higher than that with computers.

Table 4.8: Overall user satisfaction (Likert scale – 1: worst 10: best) after interaction session in each of the two tasks (standard deviation). In "Task A", participants conducted an interaction session with agent feedback first, and that without agent feedback later. In "Task B", participants conducted the interaction in the reverse order (without agent feedback first, with agent feedback later). Paired-Samples T Test shows that there is a significant difference in user satisfactions of two interaction sessions in "Task B" (5% level)).

	Task A	Task B
First session	with:7.0 (1.1)	without:5.25 (1.7)
Second session	without:6.0 (1.93)	with:7.25 (1.3)
Statistical significance	p = 0.264	p = 0.041

In the other four questionnaires, we measured (after interaction sessions) user opinions along multiple levels, such as the system performance, user satisfaction and usefulness of agent feedback.

General user feeling (1: not at all 10: very much, standard deviation) about the interface of SpeechLinks indicates that the interface is intuitive (8.71(1.3)) and easy to learn (8.18(1.1)) but not foolproof (3.5(1.0)).

To measure the effect of agent feedback, the comparison of user satisfaction between the interaction session with agent feedback and the interaction session without agent feedback is presented in Table 4.8. In addition, this comparison was conducted separately in each of the two tasks. Higher user satisfaction was observed in the interaction session with agent feedback across the two tasks. More specifically, to find out statistical significance, a Paired Sample T-test was performed on each Task and we acquired p values, 0.264 from "Task A", and 0.041 from "Task B". The observed significance level of the "Task B" confirms the statistical difference between two interaction sessions ($p < 0.05$).

Basic statistics collected from the questionnaires which support the results of Table 4.8 are the following. Overall, user feeling about the usefulness (1: not at all 10: completely) of agent feedback was 6.5 (2.4) in "Task A" and 7.4 (1.7) in "Task B". The average number of triggered agent feedback per session was 7.1 (5.0) in "Task A" and 7.9 (3.6) in "Task B". The distraction levels (1: not at all 10: completely) of agent feedback in the two tasks were 1.4 (1.3) and 1.7 (1.1) respectively. The topic difficulties (1: difficult 10: easy) in "Task A" and "Task B" were 5.7 (1.8) and 5.3 (1.4) respectively. User retry tendency (1: not at all 10: completely) in "Task A" was 6.8 (1.5) and that in "Task B" was 6.2 (2.1).

4.3.2.4 Objective Measure: Log Data Analysis

In this section, we investigated user behaviors accommodating to errors, and effects of agent feedback on the interaction efficiency. Before presenting the results, it may be interesting to note some statistics collected from the two types of interaction sessions – with/without agent feedback. Averages (with standard deviation in the parenthesis) of session dialogue time were 33 minute and 36 seconds (3 minute and 2 seconds) with agent feedback, and 32 minute and 27 seconds (4 minute and 13 seconds) without agent feedback. Averages of the number of utterances in both sessions were 77.2 (26.6), and 70.0 (19.0), respectively. Averages of utterance length (in words) were 5.3 (1.5), and 4.6 (1.2), and averages of lasting time of each utterance (in seconds) were 4.2 (0.59),

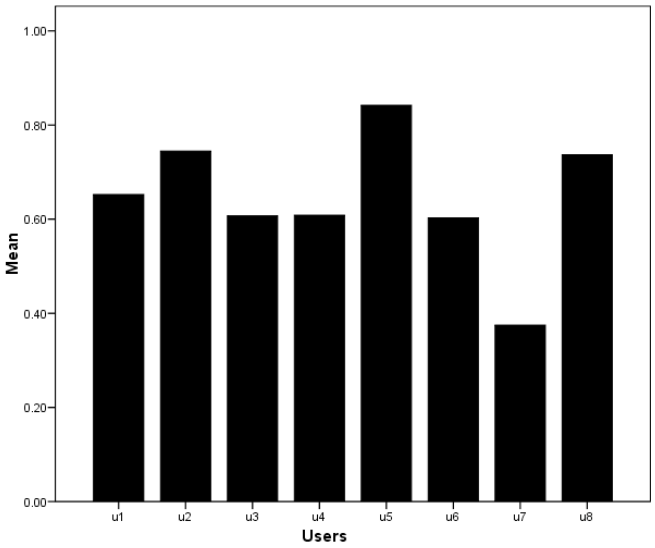


Figure 4.16: User retry rates over the interaction sessions when the ASR performance is low. Interaction sessions without agent feedback were investigated. Seven users were observed as picky and one as accommodating.

and 4.1 (0.37), respectively. Finally, overall number of triggering agent feedback in an interaction session was 10.7 (7.87) – excluding the interaction sessions without agent feedback.

In the video analysis results, we observed that on average only one participant was overall of the accommodating type, who endured relatively more recognition errors compared to the other 7 participants. In the log data analysis, we investigated retry rates of the participants under low system performance, and 7 users were observed as the picky type, and one user was the accommodating type (same as in the video analysis). The low quality (LQ) system performance is the region of low ASR confidence level. We investigated the interaction sessions without agent feedback for this analysis. The user retry rates over the interaction sessions are presented in Figure 4.16.

One of the hypotheses in using agent feedback was whether we could increase the smoothness of the interaction. This interaction efficiency is highly correlated with the time the users behave in the normal rather than picky or accommodating type regions. Normal type users are deemed to be not being in the extreme to accept/reject system errors so we expect to avoid extreme cases (such as severe repetitions or translation of large system errors) in their interactions. Intuitively, we assume smooth conversations when we the participants are behaving more “normal”. In our analysis of the data, the normal user type was exhibited more during interaction sessions with agent feedback than during interaction sessions without agent feedback as shown in Table 4.9.

Another interesting aspect is to investigate the effect of agent feedback in improving user behaviors, and in contributing to the efficiency of interactions. The agent feedback can be presented to users before the users catch the chain of same error situations. In this way, users can escape from the chain of possible error situations easily. Note that it is dependent on users to accept agent feedback, and to use alternative strategies to recover from error situations. To illustrate the effect

Table 4.9: Percentage (with standard deviation in parenthesis) of normal user type that appeared during the two interaction sessions: with/without agent feedback. More normal user type during the interaction sessions indicates more efficient interactions.

without agent feedback	with agent feedback
0.37 (0.14)	0.44 (0.14)

Table 4.10: Percentages of user behavioral change from the previous turn under possible chain of errors during the interaction sessions without/with agent feedback. The changes of user behavior (accept/retry) were counted only when the dynamic *Bayesian* reasoning identified two extreme user types (picky and accommodating) during the interaction session. Note that two extreme user types were identified internally during the interaction session without agent feedback, and user behaviors were observed at this point.

without agent feedback	with agent feedback
0.31 (0.21)	0.40 (0.16)

of agent feedback in this regard, we compared the percentages of user behavioral change from the previous turn during the interaction session without agent feedback, and during the interaction session with agent feedback (Table 4.10). In this result, the user behavioral changes were counted only when the dynamic *Bayesian* reasoning identified two extreme user types (picky and accommodating) during the interaction session. In the interaction session without agent feedback, we triggered the agent feedback internally and observed user behavior whether it was changed from the previous turn or not. Note that there is a possible chain of errors when the two extreme user types were triggered by the dynamic *Bayesian* reasoning. As shown in Table 4.10, users changed their behaviors more with the help of agent feedback onscreen, indicating that the users had more chances to escape from a chain of error situations.

4.4 Discussion and Conclusions

This chapter addressed user behavior modeling approaches in a machine-mediated setting involving bidirectional speech translation. Specifically, usability data from doctor-patient dialogs involving a two way English-Persian speech translation system was analyzed to understand two specific user behaviors. In addition to offline modeling results, data from an online experiment with agent feedback was performed and subjective and objective performance measures were reported.

We modeled user behavior with 3 user types, *Accommodating*, *Normal* and *Picky*. The granularity of user type can be adjusted according to the desired application. For example, classifying users in two categories, such as *Picky* and *Normal*, may work better when we do not want to take any steps for the case the users are extremely tolerant of errors. In the offline data, we showed that one of 3 types becomes obvious as a user keeps his/her consistent behavior under the same condition belonging to a specific type. This model can be utilized for the design of an efficient error handling

mechanism; in previous research (86), a correct interpretation of user's goal(intention) was helpful in dealing with errors in human robot dialogs. Ultimately, we believe that we can improve dialog efficiency and quality, task success, and user satisfaction that are important measures of success similar to past work on the PARADISE framework (87). In the online experiment, we addressed some of these issues with agent feedback being presented to users according to the model. High user satisfaction and interaction efficiency were reported in the interaction sessions with agent feedback.

There are several challenges that still need to be considered. One of the major challenges in empirically-based user modeling study is the availability of appropriate data. It is especially important to note that it requires a huge effort to collect, process and interpret the complex data from bilingual spoken interactions such as those considered in this study. It is well known that real human dialog data are complex to analyze and due to the high degree of variance in the data, a large volume is required to create sufficiently accurate models. In terms of data size, more training data increase the accuracy of test set (88). In addition, it is often unclear how much data is needed for optimal performance and what the appropriate features are to build a user model. These issues are of critical importance especially when we attempt to model a user in a data-driven way.

In designing a mediated device, it is important to have a good understanding of the user model, thus be able to appropriately modify the communication strategies, for example by taking specific system initiative. These system initiatives must be well founded on robust user models to ensure minimal user disruption. We designed triggering agent feedback in this fashion (to be not disruptive). However, some participants in the online experiment using agent feedback commented that they needed the feedback mostly in the early interaction sessions and repetitive feedbacks might turn out to be disruptive. How best to exploit the user model is still not a fully explored area, especially in light of partial observations (both temporally and qualitatively) of the user actions.

In the online experiment, we assumed that word error rate (WER) of the offline experiment can be substituted by ASR confidence level. This assumption is considered acceptable widely in the speech technology community. However, it is still debatable whether, under what conditions, and with what features, we can accept this assumption.

We believe this work provides a first look and motivates future investigation of the benefits of user modeling in mediated, cross-lingual interactions. The advantages of this additional model in the system are becoming apparent, even at the infancy of speech to speech translation technologies. We believe that as the devices mature, user awareness and mixed initiative will become even more critical.

Chapter 5

Knowledge as a Constraint on Uncertainty

Knowledge as a Constraint on Uncertainty for Unsupervised Classification: A Study in Part-of-Speech Tagging

Recently there has been much successful work on new models and training methods for the unsupervised learning of natural language structure, but for many practical applications, the performance of these approaches is still prohibitively low. The high costs of annotation have led to much interest in semi-supervised methods that potentially require much less labeled data to produce quality models (89). In this chapter, we explore an additional tool for learning when annotations are scarce, but there is knowledge about the problem domain. Specifically we represent facts and beliefs about the desired model output as constraints on its training, without any explicit annotation of input data.

With limits or biases on the set of possible labels for each input, our knowledge reduces the uncertainty in the classification decision for the learner. Viewing this guidance as a distribution over label output conditioned on the input data, we may quantify and compare the effects of knowledge in terms of conditional entropy, independent of the model type or training method.

We evaluate our approach on the task of unsupervised part-of-speech tagging, using the standard Hidden Markov Model tagger (90) and integrating knowledge as virtual evidence (VE) (91; 92). We apply a range of rule sets, from basic descriptions of closed-class tags and numbers to limited tag lists of the most common words, but far short of the full tagging dictionaries often used in related work. We show improvements of up to 20 or 30 points in percentage accuracy, depending on the method of state-to-label assignment, in addition to more stable and efficient convergence across model training runs. We find too that our measure of conditional labeling entropy is strongly predictive of final model performance, within a fixed model class and training set.

Finally, we address a serious problem of evaluating an unsupervised classifier in a realistic setting, which we have not seen addressed before, namely that mapping internal model states to desired labels requires the very annotated data we are supposing is not available. Accordingly, we evaluate the data requirements for producing quality mappings, specifically what proportion of the training data needs to be annotated in order to reach a certain level of accuracy, relative to the mapping given all annotations.

In sum we specify a principled means to integrate domain knowledge into the unsupervised learning of a classifier, to quantify and predict the effects of that knowledge, and to apply it effectively in a realistic setting.

The remainder of the chapter is organized as follows. Section 5.1 discusses related work and motivation for the present approach. Section 5.2 formalizes our approach and section 5.3 describes the different levels of knowledge and constraint types we evaluate. We present experimental results in section 5.4, concluding with discussion in section 5.5.

5.1 Related Work

A major motivation of the present study was Johnson's (93) thorough examination of Expectation-Maximization (EM) and Bayesian estimators for unsupervised tagging, in particular the following conclusions of that work: (1) EM can be competitive with more sophisticated Bayesian methods, (2) greatly subject to the choice of evaluation method, but (3) to a certain extent, it is possible to compensate for EM's weakness in estimating skewed distributions by constraining the model to exclude rare events. We continue these threads by exploring knowledge-based means to improve and constrain EM, while also examining further the role of evaluation method in a true unsupervised setting.

(90) introduces the classic statistical models for unsupervised and supervised part-of-speech tagging, and the source of the basic HMM tagging models employed in this chapter and many others. (94) gives probably the most detailed analysis of the induction of syntactic categories and the different types of information that lead to reasonable annotations.

Numerous variations of the original HMM tagger have been presented; additional contextual information or different dependency structures are often involved, e.g. (95) and (96).

The considerable success of log-linear models for supervised tagging (97; 98) has recently found its way to state-of-the-art results in the unsupervised setting, by either approximating the costly partition (normalization) function of those models (99) or formulating an alternate objective function that avoids it altogether (100). The Bayesian estimators of (101) bring the trigram HMM in close reach to these models.

Our view of knowledge generating hypotheses for a model to select is in part inspired by work on the unsupervised learning of morphology, where there is often a component identifying potential affixes to be fed to the learner. The work of (102) is a representative approach.

(91) introduces the notion of virtual evidence (VE) to account for judgments about relative likelihoods that are difficult to express in terms of the probability distributions and dependencies encoded by the model. (92) shows a particularly useful application of VE that is relevant here, the integration of arbitrary external models.

5.2 Knowledge as a Constraint on Uncertainty

Our approach is to improve the performance of an unsupervised classifier by using domain knowledge to guide its learning, in particular to limit or bias the choice of output label. Intuitively, restricting this choice reduces the uncertainty of the classification task and difficulty for the learner, and if we view the output as a random variable, we can measure its uncertainty in terms of the entropy of that variable's distribution. Accordingly, we can use entropy to quantify and compare the effects of different types of knowledge on the classification task, independent of any specific model, and then assess the relationship between label uncertainty and the ultimate performance of that model.

More formally, let $\mathcal{X}$ and $\mathcal{Y}$ be sample spaces of the input data and output labels, respectively, with associated random variables X and Y defined over values $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We define a knowledge source as a mapping $\mathcal{X} \rightarrow \mathcal{Y} \times [0, \infty)$, assigning a set of weights for each output y, given

an input x, that the classifier will use in its decision. If we normalize the weights and interpret them as specifying a conditional distribution $p(y|x)$, then we may characterize the knowledge source in terms of the conditional entropy of Y given X (103):

$$H(Y|X) = \sum_x p(x) \left(\sum_y p(y|x) \log p(y|x) \right) \tag{5.1}$$

$$= E_{p(x)} \left[H(Y|X = x) \right] \tag{5.2}$$

Note that this measure accords with how we commonly speak of the classification difficulty of a corpus, in terms of the average number of possible labels per input element. In fact we desire not only few possible labels, but also a skewed distribution over them to ease our decision, reducing the entropy $H(Y|X = x)$ in Eq. 5.2 for each input x. Furthermore, we would prefer the lowest entropy for the most common values of x, as reflected in the expectation under $p(x)$. In Section 5.4 we will see that this intuition is also strongly indicative of final model performance.

5.2.1 Entropy Estimation

We now address a few practical issues that arise in estimating the label entropy for a corpus and set of constraints. Given that without any prior knowledge, all outputs are equally likely, it seems that we should simply start with a uniform distribution over labels, apply the constraint rules, and then calculate the entropy of the resulting distribution. We have not, however, addressed the important question of how to interpret X and Y, namely whether they represent individual words or sequences.

While it is natural in tagging to reason about individual tokens, we only have a stable distribution $p(y|x)$ if our constraints treat all instances of x the same, that is, we have knowledge only concerning single words and their labels. This is obviously quite limiting, and so, similar to the structure of a conditional random field (104), we let labeling constraints involve all input tokens and previous labels in the current sentence, which still allows us to move through the corpus sequentially, updating each label distribution only once. The problem with this formulation is that our entropy calculation is no longer trivial, unless we make some simplifying assumptions.

For notational convenience, we take X and Y to mean the current, single-token input and output and add a context variable C representing all other words and preceding labels in the sentence. Then our label entropy is

$$\hat{H}(Y|X,C) = \sum_x \sum_c \tilde{p}(x,c) \sum_y \tilde{p}(y|x,c) \log \tilde{p}(y|x,c) \tag{5.3}$$

$$= \sum_x \tilde{p}(x) \sum_c \tilde{p}(c|x) \hat{H}(Y|X=x, C=c) \tag{5.4}$$

and, if we use the maximum-likelihood estimate for $\tilde{p}(c|x)$, the sum over c becomes simply the average of $\hat{H}(Y|X = x, C = c)$ for each instance of x we observed. Because we are effectively summing out the context, we will continue to speak of $H(Y|X)$ in subsequent sections, though this would be technically correct only if the constraint set used no contextual information.

5.3 Constraints on Unsupervised Tagging

5.3.1 Plausible Knowledge for Unsupervised Tagging

The central goal of our work is to find practical means for improving model performance when annotated data is unavailable, and so we are careful to use knowledge of the Penn Treebank's

annotations as little as possible. In our basic constraint sets, we limit our knowledge to the basic grammar and writing conventions of English, as suggested by a native speaker, or could be found in general English resources not related to or derived from the Treebank. Only in the final, 'top words' model do we allow some minimal amount of perfect knowledge, and there only for a small subset.

That said, for the sake of evaluation, we do need to express our external knowledge in terms compatible with Treebank conventions; for example, while we may perfectly understand the various uses of the word 'to', here we need to know that it takes its own reserved tag 'TO', and is never tagged as a preposition or particle.

5.3.1.0.1 Base Lexical Constraints Our first set of rules involves basic lexical knowledge of punctuation, which has reserved and mostly unambiguous tags, along with numbers and capitalization. To handle numbers, we define an 'is-number' feature matching common number formats and allow the numerical CD tag only when that feature is true for the current word, or if the CD tag is allowed for the previous word. We force a word to be tagged CD if it matches a more specific 'has-digit' feature. We handle capitalization and the proper noun tags similarly.

5.3.1.0.2 Closed Tags In the next set we add constraints listing possible words for each closed part-of-speech tag, as found in various online English references, such as Wikipedia. We did have to align these part-of-speech lists to Treebank tags, but included only words found externally to avoid use of annotation knowledge, and there is a fair amount of omission in our rules.

5.3.1.0.3 Top Words Finally, we tested constraints over the possible tags for the most frequent words in the corpus. Like much work on unsupervised tagging, we assume perfect tag lists derived from the annotated data, but limit ourselves to only the top 100 or 200 words in the entire Treebank, a more plausible scenario then producing a complete dictionary.

5.3.2 Hard and Soft Constraints as Virtual Evidence

For an HMM and any other dynamic or static Bayesian network, it is natural to view our constraints on learning as an instance of virtual evidence (91; 92), a set of judgements external to the model.

In the case of part-of-speech tagging, we might assert that a period should always be tagged as ending punctuation, a hard constraint that allows no other hypotheses. On the other hand, we might believe that the word 'walk' is four times as likely to occur as a verb compared to a noun, a softer constraint that should bias the model toward that result. Both types of knowledge can be integrated as virtual evidence, so that during EM training of the model, expectations and probability updates are adjusted accordingly, with hard constraints leading to the immediate removal of all probability mass from prohibited events.

To implement hard constraints, we add a special, binary node to the network, whose parents are the variables involved in the constraint, and whose value is always a fixed, observed value, say one.[1] The constraint node is given a deterministic conditional probability table (CPT) that defines the actual constraints (implemented by a decision tree), returning 1 only if the values of the parent variables do not violate them, and otherwise the joint state of the network is discarded as an impossible event.

Our implementation of soft constraints is similar, except now the constraint node is not observed, but has its own child, which serves as a scaling node. The scaling node itself has an observed value

[1] We thank Chris Bartels for suggesting this structure.

of one, but uses a normal, fixed CPT that assigns a certain likelihood to its parent having value one, i.e. the soft constraints being satisfied. For example, to apply constraints with a 4:1 likelihood, for constraint node c and scale node s, we define $p(s=1|c=0,1)=\{0.2, 0.8\}$.

5.4 Experimental Results

5.4.1 Experimental Details

In our experiments we performed unsupervised tagging using the standard HMM tagger introduced by (90), with first- and second-order models, implemented using the Graphical Models Toolkit (GMTK) (105). For training and evaluation, we used the Wall St. Journal portion of the Penn Treebank, version 3 (106), with data sets containing the 48k, 96k, and 193k words following the start of section 2. Due to resource constraints, we evaluated the trigram model with only the reduced, 'coarse' tag set used by (100) and (101).[2]

We repeated each experiment 10 times, training EM for 500 iterations, with parameters initialized by small, random perturbations of the uniform distribution. Because our constraints cause no changes to the model's parameter set, it was possible to use the same random initializations across constraint sets for each combination of model type, state size, and data set (e.g. all 45-state bigram models trained on 48k), and thus attempt to control for any bias from particularly good or bad initialization points.

For evaluation, we used the 'many-to-one' and 'one-to-one' labeling procedures as described by (93), which greedily assign each Viterbi state to the annotated tag with which it occurs most often, respectively either allowing or prohibiting multiple states to map to a single tag. While, as (93) and (108) mention, we may cheat with the many-to-one labeling by inflating the number of model states, this flaw seems less critical if the state count does not exceed the size of the tag set significantly.

5.4.2 Results and Analysis

Experimental results are summarized in Table 5.1, showing the performance of bigram and trigram HMM taggers with varying state sizes and increasing levels of knowledge constraints, trained and evaluated on the different data sets. The most important result is that even quite minimal sets of constraints can lead to major improvements in performance, across models and data sets, and, despite the use of simplistic tagging models, our best results reach levels of accuracy useful for practical applications.

As we might expect, the effects are most pronounced on the smaller data sets, where the constraints serve as a strong prior compensating for lack of evidence, similar to what we see with the Bayesian models of (101). The effect on both the many-to-one and one-to-one label assignments is roughly equal across experiments, so that the difference in accuracy between the two assignments changes little as we add constraints.

Following (93), we tried reducing the model state count below the number of tags, to see if the performance under the one-to-one labeling method still improved as our constraints were applied. Because a 25-state model can no longer represent all 45 tags in the corpus, both labeling methods must discard twenty tags in their assignments (possibly more with many-to-one), and so we altered our constraints to retain only those rules involving the 25 most common tags across the entire

[2]The reduced set is defined by (107, p195).

Model	48k			96k			193k		
	$H(Y\|X)$	N:1	1:1	$H(Y\|X)$	N:1	1:1	$H(Y\|X)$	N:1	1:1
Bigram (45)	5.49	33.8 (3.7)	21.7 (2.8)	5.49	42.9 (4.4)	30.1 (3.2)	5.49	52.1 (2.5)	34.4 (3.1)
+Lowercase	5.49	42.3 (2.2)	29.7 (2.3)	5.49	48.9 (2.4)	34.6 (2.5)	5.49	52.7 (2.3)	36.8 (1.9)
+Baselex	4.31	53.6 (0.8)	39.8 (1.9)	4.29	57.3 (0.8)	42.4 (1.6)	4.30	60.7 (0.8)	43.9 (1.7)
+Closed (2:1)	4.19	60.5 (0.6)	50.1 (0.8)	4.16	62.9 (0.9)	52.1 (0.8)	4.17	66.0 (0.8)	54.6 (1.0)
+Closed (4:1)	4.08	61.4 (1.4)	50.7 (1.0)	4.06	63.6 (0.8)	52.4 (1.0)	4.07	66.5 (0.8)	54.9 (1.0)
+Closed (8:1)	3.98	63.9 (1.0)	53.4 (0.8)	3.95	65.6 (0.8)	54.7 (0.8)	3.96	67.2 (0.5)	55.6 (0.9)
+Closed (16:1)	3.88	64.3 (0.9)	53.9 (1.0)	3.86	66.1 (0.8)	55.4 (0.7)	3.87	67.3 (0.4)	56.2 (0.7)
+Closed (hard)	3.71	64.9 (0.8)	54.3 (0.8)	3.69	66.2 (0.5)	55.5 (0.9)	3.70	67.4 (0.6)	56.4 (0.6)
+Top 100	3.49	69.2 (0.0)	57.8 (0.3)	3.47	70.1 (0.1)	58.6 (0.2)	3.48	71.0 (0.2)	59.5 (0.1)
+Top 200	3.49	**71.9** (0.1)	**60.5** (0.6)	3.47	**72.8** (0.1)	**61.7** (0.3)	3.48	**73.8** (0.1)	**62.1** (0.3)
Correlation (r^2)	-	0.936	0.928	-	0.917	0.916	-	0.926	0.904
Bigram (25)	4.64	34.5 (4.5)	25.0 (3.2)	4.64	39.9 (4.1)	30.0 (3.5)	4.64	47.5 (1.8)	36.2 (1.8)
+Lowercase	4.64	38.4 (3.2)	30.5 (2.9)	4.64	45.3 (2.2)	37.6 (2.1)	4.64	49.1 (2.4)	41.5 (2.5)
+Baselex	3.53	51.3 (1.0)	44.4 (2.0)	3.51	54.4 (1.2)	47.5 (2.5)	3.52	56.6 (1.2)	51.4 (1.9)
+Closed (hard)	2.94	54.9 (0.9)	48.6 (1.5)	2.92	63.0 (1.3)	**58.7** (1.2)	2.93	**65.2** (0.5)	**60.4** (1.1)
+Top 100	2.76	**61.6** (0.3)	**54.4** (0.6)	2.75	**63.4** (0.1)	55.9 (0.8)	2.75	64.9 (0.2)	57.4 (0.9)
Correlation (r^2)	-	0.904	0.922	-	0.909	0.890	-	0.938	0.905
Trigram (coarse)	3.91	46.6 (2.2)	27.4 (2.4)	3.91	57.9 (3.4)	38.7 (3.7)	3.91	62.0 (2.1)	40.8 (2.0)
+Lowercase	3.91	50.6 (2.2)	30.6 (1.8)	3.91	57.6 (2.5)	36.5 (3.2)	3.91	64.6 (1.9)	43.0 (3.4)
+Baselex	2.86	52.9 (2.3)	35.9 (3.5)	2.84	55.8 (2.0)	37.6 (3.0)	2.85	58.2 (2.3)	37.1 (2.5)
+Closed (hard)	1.76	72.0 (1.0)	53.5 (2.5)	1.75	73.8 (1.0)	57.2 (1.7)	1.76	75.9 (1.0)	59.2 (1.7)
+Top 100	1.65	**75.9** (0.1)	**57.5** (0.1)	1.64	**77.4** (0.2)	**59.0** (0.3)	1.65	**79.3** (0.2)	**61.4** (1.0)
Correlation (r^2)	-	0.895	0.919	-	0.735	0.785	-	0.609	0.664

Table 5.1: Accuracy of bigram and trigram HMM taggers trained with varying state counts (45 and 25 for full Penn tag set, and 15 for coarse tags) and increasing levels of knowledge constraints, on data sets with 48k, 96k, and 193k words. Correlation is measured between $H(Y|X)$ and accuracy of all training runs for constraint set within each model/state size/data configuration. See Section 5.3.1 for descriptions of the constraint sets.

Treebank. The results in Table 5.1 show that our constraints still have a significant benefit to model accuracy, but while the unconstrained models do exceed the one-to-one performance of the 45-state models, the improvements from added constraints do not keep pace and are inferior under both labeling methods. In part we may explain this as a result of the reduced constraint sets–for example, seven of the 'Top 100' words had no top-25 tagging, and so were removed. We also suspect that the benefits of state reduction, a fairly extreme strategy to avoid the pitfalls of rare events, are most pronounced when model performance is quite low, and, as the knowledge constraints improve accuracy, those benefits are diminished.

While increased knowledge clearly benefits model performance in these experiments, it is useful as well to determine the strength of this relationship and whether there is a correlation between accuracy and the level of constraint, as measured by the entropy $H(Y|X)$. We note first that entropy comparisons are technically valid only within a single model type (e.g. bigram HMM) and for a fixed label set and input corpus, because the calculation is independent of the model type and the cardinalities of X and Y may not vary.[3] Accordingly, within each group of experiments, we computed the Pearson correlation coefficient r^2 between the label entropy and the accuracy of the two labeling methods, as shown Table 5.1 and Figure 5.1. We found a high degree of correlation for both the 45- and 25-state bigram models, across data sets, but less so for the coarse-tag trigram models, caused mainly by a large degradation in the performance of the base lexical constraint set.

We were unable to complete a full experimental evaluation of bigram models using the coarse tag

[3]Similarly we should not compare values derived from the different vocabularies of original and lowercased input. The entropies are identical for the base and lowercased models only because, without any knowledge, Y is independent of X and thus $H(Y|X) = H(Y)$.

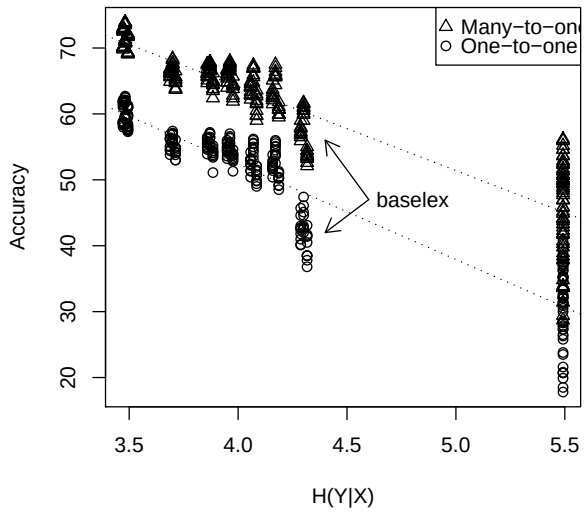


Figure 5.1: Conditional entropy $H(Y|X)$ of all constraint sets vs. accuracy, for all runs of 45-state bigram, 193k data set.

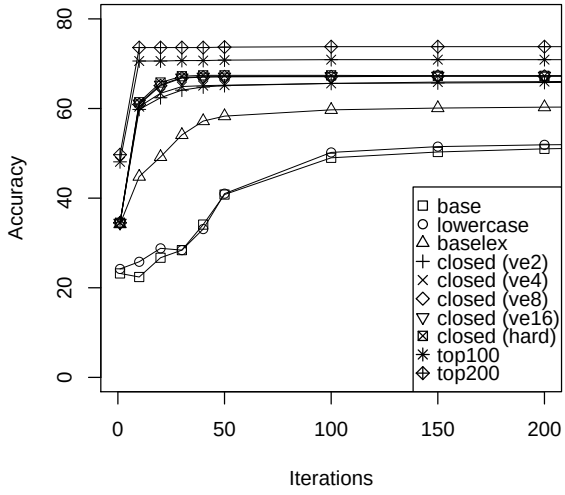


Figure 5.2: Mean accuracy for constraint sets over training iterations (only minor increases after 200), for 45-state bigram, 193k data set.

set, but we did confirm a similar degradation of the base lexical set on the 193k data.[4] Inspection of the output shows that, where some constraints had been extremely effective on the full Penn tag set, as the affected tags were lumped together with others in the coarse set, the model was led to incorrect conclusions about the more general tag. For example, numbers taking the CD tag were now mapped to a fairly heterogeneous ADJ tag with adjectives and adverbs, open-class tags for which we provided no constraints. The addition of the closed-tag constraint set, with knowledge both about members of this class (e.g. possessives) and surrounding context (e.g. determiners), filled in much incomplete knowledge and led to a large jump in both overall performance and that of ADJ (recall rose from 0.295 to 0.660). Thus, while entropy is a strong indicator of performance, there are a complex set of other factors involved.

Of course, knowledge is helpful only if the excluded hypotheses are the wrong ones. We explored the effects of imperfect knowledge by applying our closed-tag rules set as a hard constraint and then as a soft constraint with relative likelihood values ranging from 2:1 to 16:1. Because these rules are incomplete for most of the tags covered, we expected the hard constraints to perform worse, as omitted words were excluded from their correct labels. As Table 5.1 shows, however, the opposite was true, with the performance of the soft constraints suffering until we 'hardened' them with high likelihood ratios. It appears that, while the hard rules forced errors, the most common words in each tag were covered fairly well by the grammar lists we used (perhaps not too surprisingly), and the extra reduction in uncertainty outweighed the more obscure errors. A similar effect is observed in (95), where they find quite significant gains by filtering out the rare tags of each word. This does not mean necessarily that only hard constraints are useful, but it seems they can be beneficial even when they oversimplify the facts, especially for a simple model that has little hope of labeling rare and difficult events correctly. We assume, too, that it would be more ideal to separate rules according to our confidence, and assign weights accordingly.

[4]Mean performance of the 193k, 25-state bigram on coarse lowercase, baselex, closed, and dict100: 63.3/40.3, 58.2/40.2, 75.2/60.8, and 78.2/58.7.

Finally we also found that increased knowledge constraints lead to a reduction in the variance of model performance across runs, a major benefit given the problems of local extrema in unsupervised methods and the difficulty of choosing an optimal model without annotated data. For our most constrained 'Top 100' and 'Top 200' model sets, the standard deviation of the accuracy was generally under 0.5 percentage points. Additional knowledge also constrained the training process, with accuracy converging in fewer iterations. Figure 5.2 plots the accuracy for 45-state bigram models trained on the 193k data set, illustrating how the addition of rules leads to a steeper optimization surface for EM.

5.4.3 Labeling and Annotation

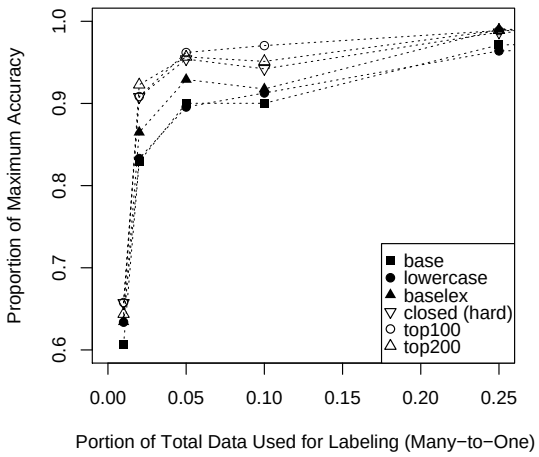


Figure 5.3: Accuracy convergence of many-to-one labeling methods, as increasing portions of the training data annotations are used to make label assignments, for 45-state bigram models trained on the 193k corpus.

There is a serious problem, however, in the applicability of the above results and those of any other unsupervised classifier: A successful mapping of the model's internal labels or states to the desired external labels requires considerable knowledge of how they best correspond. That is, to even use an unsupervised classifier we need annotated data!

To explore this issue, we labeled the output of different 45-state bigram models trained on the 193k data set, but with only part of the annotated data available to generate the mappings. Figure 5.3 shows the results of the many-to-many method, plotting each data set proportion with the accuracy of the induced label mapping, relative to the best accuracy when all data was used.[5] As an example, consider the case of the unconstrained, lowercased model. With 10% of the data, or roughly 19k words, the labeling accuracy was 48.1, compared to 52.7 when the entire set is used, so that this partial-data assignment performs at 0.91 of its full accuracy level.

While our first impression is that labeling performance converges relatively quickly, we should note that even the 5% portion represents nearly 10,000 words of annotation. Still, with the more

[5]One-to-one convergence was slightly faster, but the relative rates of the different constraints were similar.

constrained knowledge sets, 90% of optimal accuracy is reached with only 2% of the data (4k words), so once again the use of prior knowledge is extremely beneficial in a practical setting.

5.5 Discussion and Future Work

We have shown that limited amounts of domain knowledge can lead to significant performance improvements in unsupervised tagging models, with more rapid and stable convergence during training, bringing even the simple HMM tagger to levels of accuracy feasible for real application. We have motivated the analysis of knowledge as a limit on the uncertainty of a learning task, quantified by entropic measures that are independent of the statistical model and desired annotations, yet still quite predictive of final performance. Finally we have shown how, in a practical setting, knowledge helps produce a quality mapping of model states to labels, with greatly reduced requirements for annotated data.

While tagging is obviously one of the simplest NLP problems, and the HMM tagger is extremely impoverished, lacking sufficient parameters and dependencies to model many relationships, there are still many practical lessons in these results. First, despite the recent proliferation of electronic resources for lower density languages, there is still much need for unsupervised classification tasks on smaller data sets, where we might expect comparable magnitudes of improvement. Though the effects of any prior knowledge can be expected to diminish as data and model complexity grow (see (101)), the stability of learning that this knowledge provides is an important benefit, given the local extrema of unsupervised learning methods. While reducing uncertainty is not the only factor in classification performance, it is clearly significant, and entropy measures prove a useful tool in reasoning about and comparing different sets of domain knowledge.

Our current research focuses on extending these results to richer types of unsupervised classification, such a morphological analysis. We are also interested in the effects and optimal integration of domain knowledge in supervised and semi-supervised training, and with log-linear models.

Chapter 6

Robust Clustering for Speaker Diarization

Robust Agglomerative Hierarchical Clusteringfor Reliable Speaker Diarization Under Data Source Variation

A key goal for multimedia content analysis is to provide automatic description of a given multimedia datum from a semantic point of view resulting in rich meta descriptions that are more intuitive for the end user. This area of research has recently drawn much attention as a necessary step for multimedia data indexing, as demand for fast and efficient retrieval of numerous multimedia data on local or global networks increases. A fundamental step in multimedia content analysis is to segment a given multimedia datum into meaningful processing units (208)-(212), e.g., scenes. There are various segmentation methods depending upon the type of information sources (e.g., audio or video) in multimedia data and associated ways of defining a meaningful processing unit. One such method is *speaker diarization*, which divides a given datum, predominantly using speech[1], into speaker-specific segments by transcribing it in terms of "who spoke when" (213). Such speaker-specific segmentation done by speaker diarization can be beneficial to many subsequent steps in multimedia content analysis, such as for automatic speech recognition. For instance, speaker diarization enables selecting speaker-specific data that can be utilized for unsupervised speaker adaptation. It also can help provide the statistics that rely on speaker-specific information, such as frequency of speaking turn change, average speaking time per turn, number of speakers, speaking time distribution over speakers, and so on. Because of its broad significance, speaker diarization is currently regarded as one of the main categories evaluated in the Rich Transcription Evaluation led by the National Institute of Standards and Technology (NIST) (214).

As shown in Fig. 1, a speaker diarization system basically consists of three main steps following audio feature extraction. One is *speech/non-speech detection*, which separates target speech regions from the audio portions of a given multimedia stream. The others are *speaker change detection* and *speaker clustering*. Speaker change detection identifies potential speaker changing points in each speech region (or segment), and further divides the speech region (or segment) into smaller chunks of speaker-specific segments. Speaker clustering classifies speech regions (or segments) by speaker identity to append a unique label to the regions belonging to the same speaker class. These two steps can be sequentially performed either in the order mentioned, i.e., speaker change detection first and then speaker clustering, or in the opposite order. The present chapter focuses

[1]While audio and visual data can be utilized for this purpose, our specific focus in this chapter is on speech data.

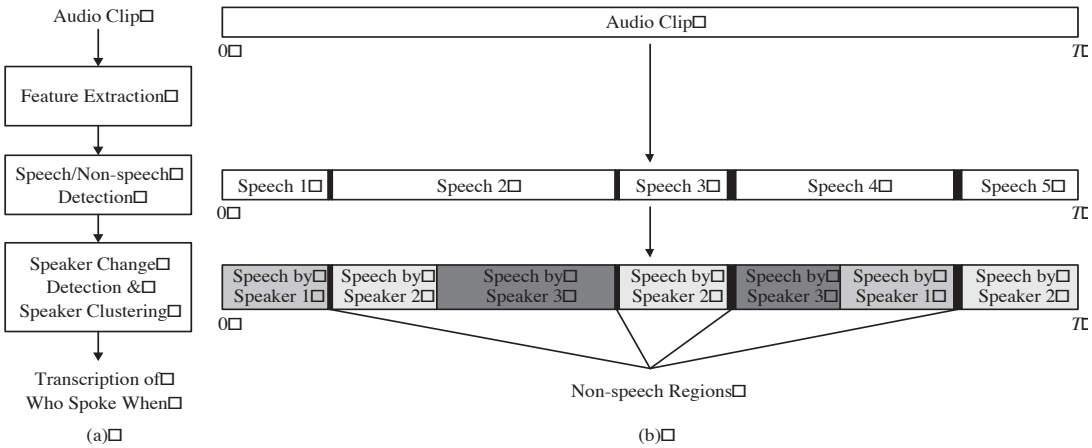


Figure 6.1: Speaker diarization: (a) Block diagram of a speaker diarization system. (b) Step-by-step graphical interpretation of how a given audio clip is transcribed (in terms of "who spoke when") by speaker diarization.

Algorithm 1 Agglomerative Hierarchical Clustering (AHC)

Require: $\{\mathbf{x}_i\}, i = 1, ..., \hat{n}$: speech segments
$\hat{C}_i, i = 1, ..., \hat{n}$: initial clusters
Ensure: $C_i, i = 1, ..., n$: finally remaining clusters
1: $\hat{C}_i \leftarrow \{\mathbf{x}_i\}$, $i = 1, ..., \hat{n}$
2: **do**
3: $i, j \leftarrow \arg\min d(\hat{C}_k, \hat{C}_l), k, l = 1, ..., \hat{n}, k \neq l$
4: merge $\hat{C}_i$ and $\hat{C}_j$
5: $\hat{n} \leftarrow \hat{n} - 1$
6: **until** diarization error rate (DER) reaches the lowest level
7: **return** $C_i, i = 1, ..., n$

on the former way, which has been more widely used in the field of speaker diarization. Under this structure for speaker change detection and speaker clustering, we further concentrate on aspects of speaker clustering, specifically, in addressing robustness issues due to data source variation in this chapter. It has been shown that data source variation causes significant performance problems in current speaker diarization systems (213),(215).

Agglomerative hierarchical clustering (AHC) (216) has been popularly used as a speaker clustering strategy in many of the speaker diarization systems that have been developed by a number of leading research groups (217)-(222), due to its simple structure and acceptable level of performance. Algorithm 1 shows how it works within the framework of speaker diarization. Using the speech segments given by the speaker change detection step as initial clusters, AHC recursively merges the closest pair of clusters until diarization error rate (DER) reaches the lowest level. In order for AHC to work properly, two critical questions need to be answered:

1. How to estimate when DER reaches the lowest level?
2. How to select homogeneous clusters in terms of speaker identity for merging at every stage of AHC so as to achieve the minimum possible level of DER?

Table 6.1: Development set of data sources. N_s: # of speakers (male:female), T_s: total speaking time (sec.), N_t: # of speaking turn changes, and T_a: average speaking time per turn (sec.). C, N, and I: data sources chosen from ICSI, NIST, and ISL meeting speech corpora respectively.

	Development Set				
	C-1	C-2	C-3	N-1	I-1
N_s	5:2	5:2	4:2	3:1	2:2
T_s	1064.9	931.3	1148.5	835.7	477.7
N_t	417	278	243	178	118
T_a	2.5	3.3	4.7	4.7	4.0

Toward addressing these questions, in the state of the art, a stopping method based on Bayesian information criterion (BIC) (223) (for the former question) and generalized likelihood ratio (GLR) as an inter-cluster distance measure (for the latter question) have been widely adopted (224)-(225).

Robustness issues in AHC are faced by both of the BIC-based stopping method and the GLR-based inter-cluster distance measurement in the presence of data source variation. Under data source variation, the BIC-based stopping method unreliably estimates the optimal stopping point where DER reaches the lowest level, while the GLR-based inter-cluster distance measurement unstably selects clusters for merging at every stage of AHC to keep the minimum possible level of DER from being achieved. In this chapter, we consider both these issues. In Section II, the data sources and the setup used for experiments in the chapter are described. The BIC-based stopping method is investigated in Section III, where we analyze the cause of its sensitivity to data source variation. In Section IV, based on the analysis in Section III, we address the robustness issue in the BIC-based stopping method by proposing a novel alternative based on information change rate (ICR). Through experiments on various meeting conversation excerpts, the ICR-based stopping method is demonstrated to be more robust to data source variation than the BIC-based one. In Section V, we also address the robustness issue in the GLR-based inter-cluster distance measurement by introducing a simple modified version of AHC, which first runs AHC with the ICR-based stopping method only on the speech segments not shorter than 3 seconds[2] in a data source and then classifies the speech segments shorter than 3 seconds into one of the clusters given by the initial AHC. This modification that we refer to as selective AHC (SAHC) is motivated by our previous analysis in (226) that the proportion of short speech segments in a data source is one significant source of variability in the minimum achievable DER across data sources. By selective classification of speech segments in terms of length, SAHC mitigates the negative effect of short speech segments on the GLR-based inter-cluster distance measurement. Finally, we conclude this chapter with comments on future work in Section VI.

[2]Let us call them long speech segments in this chapter. Accordingly, we call the speech segments shorter than 3 seconds short speech segments.

Table 6.2: Evaluation set of data sources. The notation is same as that in Table I.

	Evaluation Set									
	C-4	C-5	C-6	C-7	C-8	C-9	N-2	N-3	I-2	I-3
N_s	3:2	7:2	6:1	5:1	4:0	7:2	3:1	4:2	4:4	2:1
T_s	674.5	423.2	2336.3	1664.9	1475.9	659.7	443.4	624.1	272.4	365.3
N_t	175	129	610	531	477	158	74	143	92	72
T_a	3.8	3.3	3.8	3.1	3.1	4.1	5.9	4.3	2.9	5.0

6.1 Data Sources and Experimental Setup

Tables I and II present the development and evaluation data sets used for the experiments reported in this chapter, obtained from 15 different meeting conversation excerpts (of total length approximately 3 hours and 45 minutes). The data sources are chosen from ICSI, NIST, and ISL meeting speech corpora[3]. They are distinct from one another in terms of number of speakers (N_s), gender distribution over speakers, total speaking time (T_s), number of speaking turn changes (N_t), and average speaking time per turn (T_a). The development set will be used during parameter tuning for the stopping methods in AHC while the evaluation set will be used for performance calculation.

For the experiments presented in this chapter, we assume that both the speech/non-speech detection step and the speaker change detection step have been perfectly carried out during speaker diarization, allowing us to concentrate on the clustering issues. To enable this, we manually segmented each data source according to the reference transcription officially provided by the Linguistic Data Consortium (LDC) prior to the experiments. In order to avoid any potential confusion in performance analysis that might result from overlaps between segments, we excluded all the segments involved in any overlap during data preparation.

In order to measure DER, we used an official scoring tool, i.e., md-eval-v21.pl[4], distributed by NIST. This tool provides DER as the sum of Missed Speaker Time Rate, False Alarm Speaker Time Rate, and Speaker Error Time Rate. Due to the assumption of perfect speech/non-speech detection and speaker change detection, DER in this chapter is determined only by Speaker Error Time Rate.

Mel-frequency cepstral coefficients (MFCCs) are used as acoustic features in this chapter. Through 23 mel-scaled filter banks, a 12-dimensional MFCC vector is generated for every 20ms-long frame of speech. Every frame is shifted with the fixed rate of 10ms so that there can be an overlap between two adjacent frames.

6.2 BIC-based Stopping Method for AHC

We begin this section by providing relevant background details on GLR and BIC. The former is, as mentioned in Section I, a widely-used inter-cluster distance measure for selecting merging clusters at every stage of AHC, and the latter is a well-known model selection criterion and is utilized for the stopping method considered in this section.

[3]LDC2004S02, LDC2004S09, and LDC2004S05, respectively.

[4]This tool can be downloaded from http://www.nist.gov/speech/tests/rt/2006-spring.

6.2.1 Generalized Likelihood Ratio (GLR)

Suppose that a pair of clusters C_x and C_y are given and they consist of n-dimensional acoustic feature vectors $x = \{x_1, x_2, \cdots, x_M\}$ and $y = \{y_1, y_2, \cdots, y_N\}$, respectively. Then, GLR for the pair given is computed as follows:

$$\mathrm{GLR}\left(C_x, C_y\right) = \frac{P\left(x \cup y | H_1\right)}{P\left(x \cup y | H_2\right)}, \tag{6.1}$$

where

- H_1 (Unmerging Hypothesis): C_x and C_y are hypothesized to be left unmerged.
- H_2 (Merging Hypothesis): C_x and C_y are hypothesized to be merged so as to be a new cluster C_z, where $z = x \cup y$.

In order to mathematically calculate the two likelihoods in the right side of Eq. (1), the two hypotheses need to be modeled by probability mass or distribution functions (PMFs or PDFs) respectively. For this, single Gaussian modeling for each cluster considered (C_x and C_y for H_1, and C_z for H_2) has been popularly utilized since (225). In this chapter, we also follow this approach because single Gaussian modeling for the clusters is not only still popular in GLR computation but also much easier to be analyzed theoretically than other current modeling approaches such as Gaussian mixture modeling (GMM). Based on (225), C_x, C_y, and C_z are modeled by (multivariate) single Gaussian distributions f_X, f_Y, and f_Z with full covariance matrices respectively. Assuming that the PDFs represent random variables X, Y, and Z respectively, we can regard x, y, and z (in the modeling framework of (225)) as the sequences of independently and identically distributed (i.i.d.) random variables drawn according to the PDFs f_X, f_Y, and f_Z of random variables X, Y, and Z respectively. The mean vectors and the covariance matrices of f_X, f_Y, and f_Z are determined by way of maximizing the likelihoods of x, y, and z for f_X, f_Y, and f_Z respectively. In order words,

$$\tilde{\theta}_x = (\mu_x, \Sigma_x) = (\mu_{f_X}, \Sigma_{f_X}) = \theta_{f_X}, \tag{6.2}$$

$$\tilde{\theta}_y = (\mu_y, \Sigma_y) = (\mu_{f_Y}, \Sigma_{f_Y}) = \theta_{f_Y}, \tag{6.3}$$

and

$$\tilde{\theta}_z = (\mu_z, \Sigma_z) = (\mu_{f_Z}, \Sigma_{f_Z}) = \theta_{f_Z}, \tag{6.4}$$

where μ_x, μ_y, and μ_z are the sample mean vectors, and Σ_x, Σ_y, and Σ_z are the sample covariance matrices obtained from x, y, and z respectively. μ_{f_X}, μ_{f_Y}, and μ_{f_Z} are the mean vectors, and Σ_{f_X}, Σ_{f_Y}, and Σ_{f_Z} are the covariance matrices of f_X, f_Y, and f_Z respectively. Under this framework, Eq. (1) can be re-written as

$$\mathrm{GLR}\left(C_x, C_y\right) = \frac{p\left(x | f_X; \theta_{f_X}\right) \cdot p\left(y | f_Y; \theta_{f_Y}\right)}{p\left(z | f_Z; \theta_{f_Z}\right)} \tag{6.5}$$

$$= \frac{p\left(x | f_X; \tilde{\theta}_x\right)}{p\left(x | f_Z; \tilde{\theta}_z\right)} \cdot \frac{p\left(y | f_Y; \tilde{\theta}_y\right)}{p\left(y | f_Z; \tilde{\theta}_z\right)}. \tag{6.6}$$

We can see from Eq. (6) that GLR is always greater than or equal to 1 because both of the numerators in the equation are maximal out of the likelihoods of x and y respectively. In other words, $p(x|f_X; \tilde{\theta}_x) \geq p(x|f_Z; \tilde{\theta}_z)$ and $p(y|f_Y; \tilde{\theta}_y) \geq p(y|f_Z; \tilde{\theta}_z)$, where the equalities hold only if $C_x = C_y$ or $x = y$. This means that H_1 is always more likely than H_2, and thus GLR is not

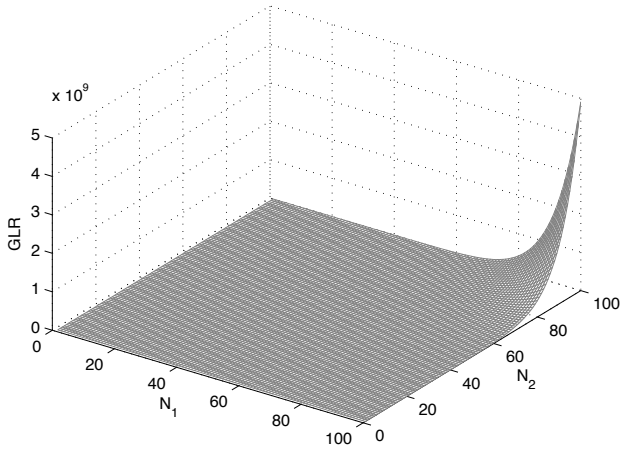


Figure 6.2: GLR for two clusters C_1 and C_2 along with the number of feature vectors in each cluster with the fixed second order statistics. $\mu_1 = 0$, $\mu_2 = 1$, and $\Sigma_1 = \Sigma_2 = 1$.

adequate to indicate that one hypothesis is more likely than the other. GLR is a measure that provides give information on how much more likely H_1 is than H_2. Therefore, the more likely H_1 is for a pair of clusters, the more distant the clusters are regarded in GLR-based distance measurement.

The drawback of GLR as a distance measure is, as mentioned in (226)-(229), that GLR tends to get larger as the total number of feature vectors within a pair of clusters under consideration increases. This can be clearly illustrated in Fig. 2, which shows GLRs between two clusters C_1 and C_2 along with the numbers of feature vectors N_1 and N_2 respectively. In order to observe the effect of the number of feature vectors, we fixed the second order statistics of $\tilde{\theta}_1$ and $\tilde{\theta}_2$ arbitrarily. (In this case, $\mu_1 = 0$, $\mu_2 = 1$, and $\Sigma_1 = \Sigma_2 = 1$.) This figure explicitly shows the exponential rising-up of GLR as the numbers of feature vectors increase. Consequently, in GLR-based inter-cluster distance measurement, a pair of homogeneous clusters consisting of a small number of feature vectors are likely to have a smaller GLR value and be regarded as mutually closer than those consisting of a large number of feature vectors. Besides, a pair of heterogeneous clusters consisting of a small number of feature vectors might have a smaller GLR value and be regarded as mutually closer than a pair of homogeneous clusters consisting of a large number of feature vectors, which is undesirable.

This undesirable tendency of GLR can be confirmed by analyzing GLR computation with a few basic concepts in the field of information theory. Let us begin this analysis with Eq. (5). We can re-write the equation as below without loss of generality by applying logarithm to both sides:

$$\begin{aligned} &\ln \mathrm{GLR}\left(C_x, C_y\right) \\ &= \ln \frac{p\left(x|f_X;\theta_{f_X}\right)\cdot p\left(y|f_Y;\theta_{f_Y}\right)}{p\left(z|f_Z;\theta_{f_Z}\right)} \\ &= \ln f_X\left(x_1, x_2, \cdots, x_M\right) + \ln f_Y\left(y_1, y_2, \cdots, y_N\right) - \\ &\quad \ln f_Z\left(x_1, \cdots, x_M, y_1, \cdots, y_N\right). \end{aligned} \tag{6.7}$$

Considering that GLR computation intrinsically assumes the weak law of large numbers[5] to be satisfied during its procedure, we can apply the asymptotic equipartition property[6] (AEP) widely-known as the consequence of the weak law of large numbers in the field of information theory to the right side term of Eq. (7). Then, the equation can be simplified to

$$\begin{aligned} \ln \mathrm{GLR}\left(C_x, C_y\right) &= -M \cdot h\left(X\right) - N \cdot h\left(Y\right) + \\ & \quad (M+N) \cdot h\left(Z\right), \end{aligned} \tag{6.9}$$

where h is entropy. Since entropy for an n-dimensional multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ can be obtained (according to (230)) as a closed form of $\frac{1}{2}\ln(2\pi e)^n|\Sigma|$ where $|\cdot|$ is determinant, we can further simplify Eq. (9) to

$$\begin{aligned} & \ln \mathrm{GLR}\left(C_x, C_y\right) \\ &= -M \cdot \frac{1}{2}\ln\left(2\pi e\right)^n \left|\Sigma_x\right| - N \cdot \frac{1}{2}\ln\left(2\pi e\right)^n \left|\Sigma_y\right| + \\ & \quad (M+N) \cdot \frac{1}{2}\ln\left(2\pi e\right)^n \left|\Sigma_z\right| \\ &= \frac{M+N}{2}\ln\left|\Sigma_z\right| - \frac{M}{2}\ln\left|\Sigma_x\right| - \frac{N}{2}\ln\left|\Sigma_y\right|, \end{aligned} \tag{6.10}$$

where Σ_z has the following relation with Σ_x and Σ_y:

$$\begin{aligned} \Sigma_z &= \frac{M \cdot \Sigma_x + N \cdot \Sigma_y}{M+N} + \frac{M \cdot \mu_x \mu_x^T + N \cdot \mu_y \mu_y^T}{M+N} - \\ & \quad \frac{M \cdot \mu_x + N \cdot \mu_y}{M+N} \cdot \left(\frac{M \cdot \mu_x + N \cdot \mu_y}{M+N}\right)^T \end{aligned} \tag{6.11}$$

because $z = x \cup y$.

Based on this, suppose that we compute GLR between two clusters $C_{x'}$ and $C_{y'}$, where x' and y' are the sequences of i.i.d. random variables drawn according to the PDFs f_X and f_Y, and their cardinalities are $2M$ and $2N$ respectively. In other words, x' (or y') has the same second order statistics with x's (or y's) but twice the number of feature vectors within x (or y). Then, we obtain the same $\Sigma_{z'}$ (where $z' = x' \cup y'$) with Σ_z using Eq. (11), and hence

$$\begin{aligned} & \ln \mathrm{GLR}\left(C_{x'}, C_{y'}\right) \\ &= (M+N)\ln\left|\Sigma_{z'}\right| - M \cdot \ln\left|\Sigma_{f_X}\right| - N \cdot \ln\left|\Sigma_{f_Y}\right| \\ &= (M+N)\ln\left|\Sigma_z\right| - M \cdot \ln\left|\Sigma_x\right| - N \cdot \ln\left|\Sigma_y\right| \\ &= 2 \cdot \ln \mathrm{GLR}\left(C_x, C_y\right), \end{aligned} \tag{6.12}$$

The above example indicates that $\ln \mathrm{GLR}$ linearly increases (or GLR exponentially increases) with the fixed second order statistics as the numbers of feature vectors within a pair of clusters under consideration get larger, which is consistent with what is shown in Fig. 2.

[5]The weak law of large numbers states that a sample mean and a sample variance converge in probability towards the expected value and the second central moment of a corresponding random variable respectively. In GLR computation, this law is inherent to Eqs. (2)-(4).

[6]Let $x_1, x_2, \cdots, x_M$ be the sequence of i.i.d. random variables drawn according to the PDF f_X of a random variable X. Then, according to (230), the AEP states that

$$-\frac{1}{M}\ln f_X\left(x_1, x_2, \cdots, x_M\right) = h\left(X\right) \text{ in probability}, \tag{6.8}$$

where h is entropy.

6.2.2 Bayesian Information Criterion (BIC)

BIC (223) was primarily intended for model (or PDF) selection, specifically for the problem of how to select the best model for given observations from candidate models. A basic model selection strategy based on BIC is as follows:

1. Compute BIC scores for all candidate models.

$$\begin{aligned} \mathrm{BIC}(f) &= \ln p(x|f;\theta_f) - \mathbf{P}_f \\ &= \ln p(x|f;\theta_f) - \frac{1}{2}\#(\theta_f)\ln M, \end{aligned} \tag{6.13}$$

   where $x = \{x_1, x_2, \cdots, x_M\}$ represents given M observations, f is a model (or PDF), θ_f is a set of model parameters for f, and $\#(\theta_f)$ is the total number of model parameters for f.

2. Select the model whose BIC score is the highest as the best one to represent the observations.

The core of BIC is that the log-likelihood of given observations for a model is penalized by $\mathbf{P}_f$, which is determined by the total number of model parameters and the logarithm of the cardinality of the observations. This prevents the model having the most number of parameters from being chosen all the time as the best one, which is a well-known issue in model selection based on maximum likelihood without penalization.

6.2.3 BIC-based Stopping Method for AHC

Keeping both GLR and BIC in mind, we now investigate the BIC-based stopping method for AHC. This conventional method to search for the optimal stopping point for AHC (when DER reaches the lowest level) was originally introduced in (224) by Chen and Gopalakrishnan. It basically stops AHC at the point when the closest pair among all pairs of remaining clusters are decided to be not homogeneous for the first time, based on the reasoning that if the closest pair of clusters were heterogeneous then so would be any other pair of clusters, and thus there would be no more need for merging in AHC. Decision of homogeneity for the closest pair of clusters at every stage of AHC is done by comparing the BIC scores of the clusters for two hypotheses of 'Unmerging' and 'Merging'. These two hypotheses are the same as those (H_1 and H_2) used in GLR computation in Section III.A, and in this case H_2 supports homogeneity while H_1 supports heterogeneity. As in GLR computation, the two clusters considered are modeled by (multivariate) single Gaussian distributions with maximum likelihood parameter estimation. The details of how the BIC-based stopping method works for AHC are as follows[7]:

1. For the closest pair of clusters C_x and C_y consisting of feature vectors $x = \{x_1, x_2, \cdots, x_M\}$

[7] We used the same notation in Section III.A for single Gaussian modeling for clusters.

and $y = \{y_1, y_2, \cdots, y_N\}$ respectively, compute the BIC scores of $x \cup y$ for H_1 and H_2.

$$
\begin{aligned}
&\mathrm{BIC}\left(H_1\right) \\
&= \ln P\left(x \cup y | H_1\right) - \lambda \cdot \mathbf{P}_{H_1} \\
&= \ln P\left(x \cup y | H_1\right) - \lambda \cdot \frac{1}{2} \# \left(H_1\right) \ln N_{total} \\
&= \ln \left\{p\left(x | f_X; \theta_{f_X}\right) \cdot p\left(y | f_Y; \theta_{f_Y}\right)\right\} - \\
&\quad \lambda \cdot \frac{1}{2} \left\{\# \left(\theta_{f_X}\right) + \# \left(\theta_{f_Y}\right)\right\} \ln N_{total} \\
&= \ln \left\{p\left(x | f_X; \tilde{\theta}_x\right) \cdot p\left(y | f_Y; \tilde{\theta}_y\right)\right\} - \\
&\quad \lambda \cdot \frac{1}{2} \left[2 \left\{n + \frac{1}{2} n\left(n+1\right)\right\}\right] \ln N_{total}.
\end{aligned} \tag{6.14}
$$

$$
\begin{aligned}
&\mathrm{BIC}\left(H_2\right) \\
&= \ln P\left(x \cup y | H_2\right) - \lambda \cdot \mathbf{P}_{H_2} \\
&= \ln P\left(x \cup y | H_2\right) - \lambda \cdot \frac{1}{2} \# \left(H_2\right) \ln N_{total} \\
&= \ln \left\{p\left(x | f_Z; \theta_{f_Z}\right) \cdot p\left(y | f_Z; \theta_{f_Z}\right)\right\} - \\
&\quad \lambda \cdot \frac{1}{2} \# \left(\theta_{f_Z}\right) \ln N_{total} \\
&= \ln \left\{p\left(x | f_Z; \tilde{\theta}_z\right) \cdot p\left(y | f_Z; \tilde{\theta}_z\right)\right\} - \\
&\quad \lambda \cdot \frac{1}{2} \left\{n + \frac{1}{2} n\left(n+1\right)\right\} \ln N_{total}.
\end{aligned} \tag{6.15}
$$

In Eqs. (16) and (17), λ is the parameter that should be tuned a priori for minimizing averaged DER with a development set of data sources (which will be explained more in detail later), N_{total} is the total number of feature vectors for the entire clusters given as an input for AHC, and n is the dimension of feature vectors.

2. Compute $\Delta\text{BIC}\left(C_x, C_y\right) = \text{BIC}\left(H_1\right) - \text{BIC}\left(H_2\right)$.

$$
\begin{aligned}
&\Delta\text{BIC}\left(C_x, C_y\right) \\
&= \ln\left\{p\left(x|f_X;\tilde{\theta}_x\right)\cdot p\left(y|f_Y;\tilde{\theta}_y\right)\right\} - \\
&\quad \lambda\cdot\frac{1}{2}\left[2\left\{n+\frac{1}{2}n\left(n+1\right)\right\}\right]\ln N_{total} - \\
&\quad \ln\left\{p\left(x|f_Z;\tilde{\theta}_z\right)\cdot p\left(y|f_Z;\tilde{\theta}_z\right)\right\} + \\
&\quad \lambda\cdot\frac{1}{2}\left\{n+\frac{1}{2}n\left(n+1\right)\right\}\ln N_{total} \\
&= \ln\frac{p\left(x|f_X;\tilde{\theta}_x\right)\cdot p\left(y|f_Y;\tilde{\theta}_y\right)}{p\left(x|f_Z;\tilde{\theta}_z\right)\cdot p\left(y|f_Z;\tilde{\theta}_z\right)} - \\
&\quad \lambda\cdot\frac{1}{2}\left\{n+\frac{1}{2}n\left(n+1\right)\right\}\ln N_{total} \\
&= \ln\text{GLR}\left(C_x, C_y\right) - \\
&\quad \lambda\cdot\frac{1}{2}\left\{n+\frac{1}{2}n\left(n+1\right)\right\}\ln N_{total} \qquad (6.16) \\
&\underset{H_2}{\overset{H_1}{\gtrless}} \quad 0.
\end{aligned}
$$

3. If $\Delta\text{BIC}\left(C_x, C_y\right) < 0$ or $\text{BIC}\left(H_1\right) < \text{BIC}\left(H_2\right)$, decide that C_x and C_y are homogeneous and merge them. Otherwise, do not merge them and stop AHC.

The stopping criterion mentioned above can be re-written as

$$
\ln\text{GLR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \lambda\cdot c\cdot\ln N_{total}, \qquad (6.17)
$$

where $c = \frac{1}{2}\left\{n+\frac{1}{2}n\left(n+1\right)\right\}$ is a constant. This criterion could be replaced by

$$
\ln\text{GLR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \lambda\cdot c\cdot\ln\left(M+N\right). \qquad (6.18)
$$

This modified criterion was introduced in (222) based on its better performance for estimating the optimal stopping point for AHC than Eq. (17). In this chapter, we will consider Eq. (18) as a baseline stopping criterion for the BIC-based stopping method for this reason. From this point on, the stopping criterion that we are mentioning throughout the chapter thus points out Eq. (18), not Eq. (17).

6.2.4 Tuning Parameter λ

An important aspect to note for this BIC-based stopping method is the use of the tuning parameter λ in Eqs. (14) and (15). This parameter is not included in the original BIC score computation as shown in Eq. (13), which means that the parameter was intentionally introduced when applying BIC to devise a stopping method for AHC. Unfortunately, there is no explicit explanation in (224)

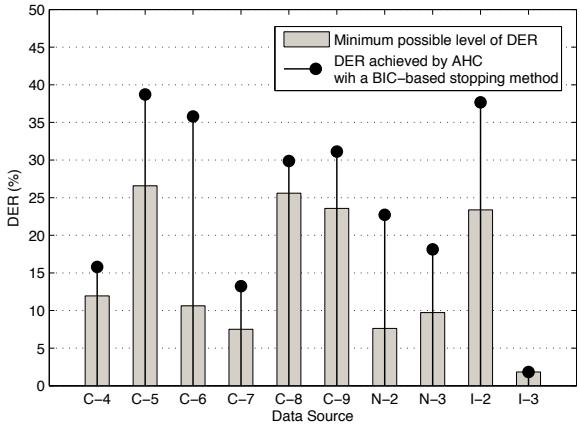


Figure 6.3: Comparison of the minimum possible levels of DERs for the evaluation data set described in Section II with the respective DERs achieved by AHC with the BIC-based stopping method with $\lambda = 12.0$. Average DER degradation by wrong estimation of the optimal stopping point is about 9.65% (absolute) per data source.

of why λ is necessary and how it can be optimally chosen. In the field of speaker diarization, however, the parameter is widely considered as a weighting factor to lift up the level of the whole right side term of Eq. (18), and is generally tuned so as for the stopping criterion to provide the minimum averaged DER for a development data set. (In this chapter, we set λ to be 12.0 because $\lambda = 12.0$ minimized averaged DER for our development data set presented in Section II.)

A problem is that λ does not work globally because it is tuned only based on a development data set. Such a tuned parameter cannot guarantee the stopping criterion to correctly estimate the optimal stopping points for data sources in a different data set, due to its dependency upon the data set used for tuning. This problem is clearly confirmed in Fig. 3[8], where comparison of the minimum possible levels of DERs for the evaluation data set described in Section II with the respective DERs achieved by AHC with the BIC-based stopping method with $\lambda = 12.0$. We can see from the figure that with $\lambda = 12.0$ the BIC-based stopping method does not reliably estimate when DER reaches the lowest level for the evaluation data set. In our experiments, the impact of incorrect estimation of the optimal stopping point is detrimental specifically for C-5, C-6, N-2, and I-2 while it is not the case for C-4, C-8, and I-3. Average DER degradation due to such incorrect estimation is about 9.65% (absolute) per data source.

In order to handle this problem, one interesting approach was proposed in (231) based on the idea of (232), which is to automatically erase λ by equalizing $\#(H_1)$ to $\#(H_2)$ in the computation of BIC scores for H_1 and H_2. For this, a Gaussian mixture model (GMM) with m model parameters for each cluster considered (C_x and C_y) for H_1 and another GMM with $2m$ model parameters for a hypothetically merged cluster (C_z) for H_2 were utilized respectively. By doing so, this approach can avoid parameter tuning. However, it has some side effects such as increased computing time for training GMMs at every stage of AHC. Moreover, the approach does not directly take care of a fundamental cause for the robustness issue of the BIC-based stopping method, which is the stopping criterion being not robust to data source variation.

[8]In this experiment, GLR was used as an inter-cluster distance measure for AHC to select the closest pair of clusters at every stage.

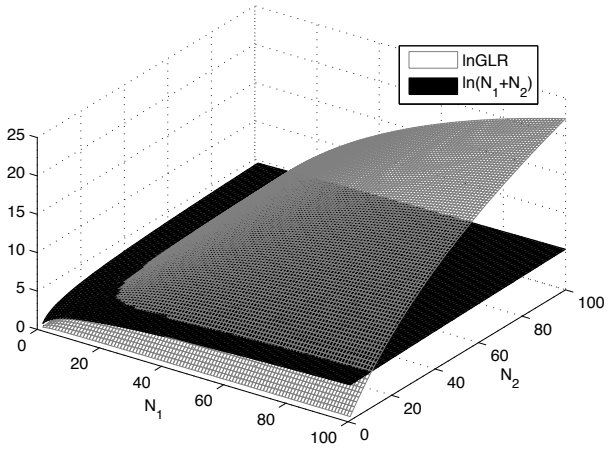


Figure 6.4: $\ln \mathrm{GLR}$ and $\ln(M+N)$ $(= \ln(N_1+N_2)$ in this case) for the same clusters considered in Fig. 2 along with the number of feature vectors in each cluster with the fixed second order statistics, $\mu_1 = 0$, $\mu_2 = 1$, and $\Sigma_1 = \Sigma_2 = 1$.

6.2.5 Sensitivity of the Stopping Criterion to Data Source Variation

The stopping criterion of the BIC-based method, Eq. (18), has an intrinsic flaw in terms of robustness to data source variation because it utilizes GLR. As aforementioned in Section III.A, GLR is sensitive to the numbers of feature vectors within the clusters considered. As a result, the left side term of Eq. (18), $\ln \mathrm{GLR}$, is affected by several aspects in the entire speech segments given as an input data source for AHC beyond just the statistical difference between the clusters considered. This is because the size of the clusters considered by the BIC-based stopping method at a certain stage of AHC is determined jointly by the total length of the segments given as an input for AHC, the distributions of the segments in length and speaker identity, and merging procedures at the previous stages of AHC. One might claim that the right side term of Eq. (18) is also affected by the numbers of feature vectors within the clusters considered due to $\ln(M+N)$, so the stopping criterion looks robust to data source variation. However, $\ln \mathrm{GLR}$ grows in a linear fashion[9] in proportion to M and N while $\ln(M+N)$ increases in a logarithmic fashion, which is well shown in Fig. 4. $\ln \mathrm{GLR}$ is fast increasing along M and N, but $\ln(M+N)$ looks relatively flat in the figure. This indicates that the right side term of Eq. (18) cannot compensate the data dependency of the left side term fully enough, and the stopping criterion is thus highly likely to vary across data sources. For this reason, it is too difficult to set a global λ.

6.3 Information Change Rate (ICR) and ICR-based Stopping Method for AHC

In the previous section, we investigated the BIC-based stopping method for AHC and underscored that a fundamental reason for the robustness issue of the method is the stopping criterion being not robust to data source variation. In this section, based on the analysis in Section III, we propose

[9] We confirmed in Section III.A that GLR exponentially increased in proportion to the numbers of feature vectors within the clusters considered.

a new stopping method for AHC that is more robust to data source variation than the BIC-based one.

6.3.1 Information Change Rate (ICR)

First, we propose a new statistical distance measure between clusters, *information change rate* (ICR), which is defined as follows for a pair of clusters C_x and C_y consisting of feature vectors $x = \{x_1, x_2, \cdots, x_M\}$ and $y = \{y_1, y_2, \cdots, y_N\}$, respectively:

$$\mathrm{ICR}\left(C_x, C_y\right) \triangleq \frac{1}{M+N} \ln \mathrm{GLR}\left(C_x, C_y\right). \tag{6.19}$$

In short, ICR is the normalized version of $\ln \mathrm{GLR}$. This simple idea of normalizing $\ln \mathrm{GLR}$ with the total number of feature vectors within a pair of clusters under consideration was inspired by analyzing GLR with an information-theoretic perspective. Let us consider Eq. (9) in Section III.A again. Considering that entropy can be regarded as average description length for a random sample from a given PDF, we can separate the right side term of the equation into the following two parts:

$$\begin{aligned} \ln \mathrm{GLR}\left(C_x, C_y\right) = & \underbrace{(M+N) \cdot h\left(Z\right)}_{\text{Total description length for } z=x \cup y \text{ under } H_2} - \\ & \underbrace{\left\{M \cdot h\left(X\right) + N \cdot h\left(Y\right)\right\}}_{\text{Total description length for } z \text{ under } H_1}. \end{aligned} \tag{6.20}$$

This means that $\ln \mathrm{GLR}$ equals difference between the *total description lengths* for the whole feature vectors considered under the two hypotheses H_1 (Unmerging) and H_2 (Merging). That is, $\ln \mathrm{GLR}$ represents how much amount of information would be *totally* changed by merging the clusters considered. Thus, it is natural to expect that a certain distance measure, if it represents how much amount of information would be changed *on average* over feature vectors by merging the clusters considered, could avoid being affected by the size of the clusters. ICR satisfies such an expectation, which is the reason why we named our proposed distance measure information change rate. From Eqs. (19) and (20), we can obtain a different version of ICR as follows:

$$\mathrm{ICR}\left(C_x, C_y\right) = h\left(Z\right) - \frac{M \cdot h\left(X\right) + N \cdot h\left(Y\right)}{M+N}. \tag{6.21}$$

Let us consider how ICR is expressed for two extreme examples:

- Ex 1: $C_x = C_y$ or $x = y$.

$$\begin{aligned} \mathrm{ICR}\left(C_x, C_y\right) &= \mathrm{ICR}\left(C_x, C_y\right) \\ &= h\left(X\right) - \frac{M \cdot h\left(X\right) + M \cdot h\left(X\right)}{M+M} \\ &= h\left(X\right) - h\left(X\right) \\ &= 0 \end{aligned}$$

Table 6.3: Comparison of ICR with other measures utilizing the idea of normalizing GLR. C_x and C_y: two clusters consisting of M and N feature vectors respectively, α: parameter empirically determined, and n: dimension of feature vectors.

ICR (C_x, C_y)	PLR in (228)	NLLR in (229)
$\frac{1}{M+N} \ln \mathrm{GLR}(C_x, C_y)$	$\frac{1}{(M+N)^\alpha} \mathrm{GLR}(C_x, C_y)$	$\frac{1}{(M+N)\cdot n} \ln \mathrm{GLR}(C_x, C_y)$

- Ex 2: C_x and C_y are mutually independent.

$$\begin{aligned}
&\mathrm{ICR}(C_x, C_y) \\
&= h(X) + h(Y) - \frac{M \cdot h(X) + N \cdot h(Y)}{M+N} \\
&= \frac{(M+N) \cdot h(X) + (M+N) \cdot h(Y)}{M+N} - \\
&\quad \frac{M \cdot h(X) + N \cdot h(Y)}{M+N} \\
&= \frac{N \cdot h(X) + M \cdot h(Y)}{M+N}
\end{aligned}$$

6.3.2 Comparison of ICR with other ICR-like inter-cluster distance measures

In fact, there have been several ICR-like inter-cluster distance measures to normalize GLR in the field of speaker diarization, specifically for speaker change detection. Table III compares two of such measures, i.e. penalized likelihood ratio (PLR) (228) and normalized log-likelihood ratio (NLLR) (229), with ICR. PLR normalizes GLR with the α-th power of the sum of feature vectors within the clusters considered. However, it does not appear promising in terms of mitigating the effect of cluster size on distance measurement, because

$$\ln \mathrm{PLR}(C_x, C_y) = \ln \mathrm{GLR}(C_x, C_y) - \alpha \cdot \ln(M+N). \tag{6.22}$$

As shown in Section III.E, $\ln(M+N)$ cannot compensate the dependency of $\ln \mathrm{GLR}$ on cluster size entirely. Thus, it is difficult to set a global α. On the other hand, NLLR is very similar to ICR and its relation to ICR is shown as follows:

$$\mathrm{NLLR}(C_x, C_y) = \frac{1}{n} \mathrm{ICR}(C_x, C_y). \tag{6.23}$$

But it has a different physical meaning from that of ICR because it further normalizes $\ln \mathrm{GLR}$ with the dimension of feature vectors.

6.3.3 ICR as a measure to decide homogeneity for clusters

Since ICR represents how much amount of information would be changed on average over feature vectors by merging the clusters considered, it is natural to expect ICR to be very small when the

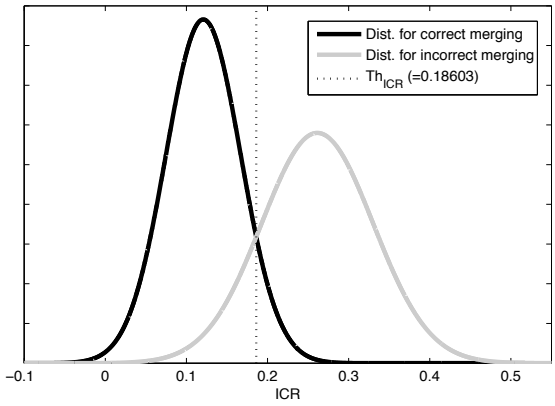


Figure 6.5: Distributions for correct and incorrect merging in terms of ICR. The threshold η is set so as to minimize classification error between the two distributions. All the merging processes used for obtaining the distributions were picked up from our development data set, and they corresponded to more than 30 seconds.

clusters considered are homogeneous in terms of speaker identity and each cluster is large enough to fully cover the intra-speaker variance of corresponding speaker identity. In other words, ICR will be small when the clusters considered have the same speaker identity source and do not need additional information for representing full speaker characteristics. On the contrary, ICR will be relatively large when the clusters considered are heterogeneous, or when they are homogeneous but contain small feature vectors to cover only a part of the speaker characteristics. Thus, ICR could properly work as a measure to decide homogeneity for clusters if every cluster considered were large enough to fully represent the characteristics of the corresponding speaker identity. In this chapter, we assume that a cluster containing feature vectors which correspond to more than 30 seconds is such a large enough cluster. This assumption is based on the fact that it requires long speech utterances (at least longer than 20 seconds) to derive reliable speaker characteristics (233)-(235).

Fig. 5 displays distributions for merging processes between homogeneous clusters (or correct merging) and for those between heterogeneous clusters (or incorrect merging) in terms of ICR. The distributions were assumed to be Gaussian, and their sample means and sample variances were respectively obtained based on the ICR values of the merging processes picked up from AHC procedures for our development data set. All the merging processes we selected occurred between the clusters corresponding to more than 30 seconds. Using the distributions in the figure, we set a threshold $\eta = \mathrm{Th}_{\mathrm{ICR}}$ to be 0.18603, with which classification error between the two distributions can be minimized. In this chapter, we thus regard a pair of clusters having ICR less than $\eta = 0.18603$ as homogeneous in terms of speaker identity.

6.3.4 ICR-based Stopping Method for AHC

Based on ICR and its applicability to inter-cluster homogeneity decision in terms of speaker identity, we now introduce an ICR-based stopping method for AHC. This method is distinct from the BIC-based one in terms of 1) stopping criterion and 2) the order of the clusters considered. Its details are as follows:

Table 6.4: ICR-based stopping method vs. BIC-based stopping method. $c = \frac{1}{2}\left\{n + \frac{1}{2}n\left(n+1\right)\right\}$, where n is the dimension of feature vectors. $n = 12$, $\eta = 0.18603$, and $\lambda = 12.0$ in this chapter.

	ICR-based Stopping Method	BIC-based Stopping Method
Criterion	$\mathrm{ICR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \eta$	$\ln \mathrm{GLR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \lambda \cdot c \cdot \ln\left(M+N\right)$
Right side term in criterion	Fixed during AHC	Floating along with M and N during AHC
Computational complexity for criterion	Complexity for computing $\ln \mathrm{GLR}\left(C_x, C_y\right)$ and $\eta \cdot (M+N)$	Complexity for computing $\ln \mathrm{GLR}\left(C_x, C_y\right)$ and $\lambda \cdot c \cdot \ln\left(M+N\right)$
Order of clusters considered	From the pair of clusters merged at the last stage of AHC	From the pair of clusters merged at the first stage of AHC

1. Wait until AHC reaches the end of its merging processes, i.e., wait until all the clusters given for AHC are merged to one cluster.
2. For the pair of clusters merged at the last stage of AHC, C_x and C_y, consisting of feature vectors $x = \{x_1, x_2, \cdots, x_M\}$ and $y = \{y_1, y_2, \cdots, y_N\}$ respectively, compute ICR.
3. Compare ICR with η:

$$\mathrm{ICR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \eta. \tag{6.24}$$

   If $\mathrm{ICR}\left(C_x, C_y\right) > \eta$, decide that C_x and C_y are heterogeneous in terms of speaker identity and consider the pair of clusters merged at the next latest stage of AHC. Otherwise, stop considering more merging processes and select the stage previously considered as the stopping point.

The ICR-based stopping method depends upon the reasoning[10] that all merging processes during AHC after the optimal stopping point would occur between heterogeneous clusters. The reason why this stopping method starts its consideration from the pair of clusters merged at the last stage of AHC is because such a strategy can make the stopping criterion Eq. (24) consider large clusters only. As mentioned in the previous subsection, ICR can properly work as a homogeneity decision measure only for large enough clusters to represent full speaker characteristics respectively. Eq. (24) can be re-written as follows:

$$\ln \mathrm{GLR}\left(C_x, C_y\right) \underset{H_2}{\overset{H_1}{\gtrless}} \eta \cdot \left(M+N\right). \tag{6.25}$$

Comparing this criterion with Eq. (18) for the BIC-based stopping method, we can see that the difference of computational complexity between the two stopping methods is thus negligible. For easier understanding of the ICR-based stopping method for AHC, Table IV is presented.

[10]The BIC-based stopping method for AHC also relies on the same reasoning.

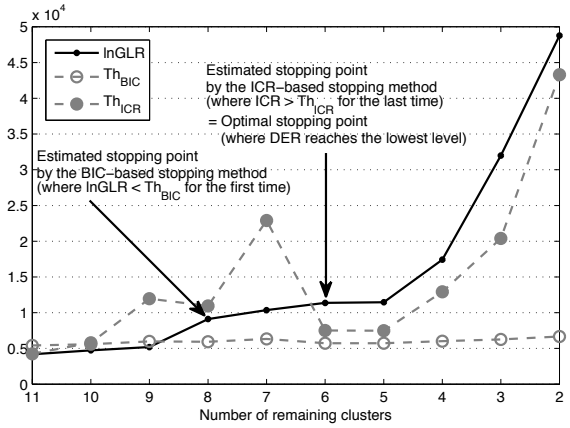


Figure 6.6: $\ln \mathrm{GLR}$, $\mathrm{Th}_{\mathrm{BIC}} = \lambda \cdot c \cdot \ln(M+N)$, and $\mathrm{Th}_{\mathrm{ICR}} = \eta \cdot (M+N)$ for C-6, where $\lambda = 12.0$ and $\eta = 0.18603$. The stopping point estimated by the ICR-based stopping method is identical to the optimal one in this case.

Fig. 6 shows $\ln \mathrm{GLR}$, $\mathrm{Th}_{\mathrm{BIC}} = \lambda \cdot c \cdot \ln(M+N)$, and $\mathrm{Th}_{\mathrm{ICR}} = \eta \cdot (M+N)$ for the data source C-6 in our evaluation data set, where $\lambda = 12.0$ and $\eta = 0.18603$. This figure focuses on the variations of the three terms at the final 10 merging processes during AHC for C-6. From the figure, we can see that $\mathrm{Th}_{\mathrm{ICR}}$ varies along with $\ln \mathrm{GLR}$ while $\mathrm{Th}_{\mathrm{BIC}}$ does not. The observation that $\mathrm{Th}_{\mathrm{BIC}}$ looks almost flat compared to $\ln \mathrm{GLR}$ is consistent with what was shown in Fig. 4 in Section III.E, and verifies that Eq. (18) is not robust to data source variation. In contrast, the robustness of the criterion in Eq. (24) or Eq. (25) to data source variation is demonstrated through the figure above.

Fig. 7[11] presents AHC performance using the ICR-based stopping method ($\eta = 0.18603$) for the evaluation data set. In the figure, we can observe that the proposed stopping method exactly detected the optimal stopping points for all the data sources except C-4, C-8, and C-9. Even for the three data sources, gaps between DERs at the estimated stopping points and those at the optimal ones are shown to be insignificant. Compared to the results obtained using AHC with the BIC-based stopping method for the same data set (shown in Fig. 3), the results in this figure are much improved overall, and indicate that the ICR-based stopping method is superior to the BIC-based one in terms of robustness to data source variation. Consequently, the ICR-based stopping method for AHC led to average DER improvement by 8.76% (absolute) and 35.77% (relative) per data source, compared to the conventional BIC-based one.

6.4 Selective Agglomerative Hierarchical Clustering (SAHC)

In this section, we tackle the robustness issue of inter-cluster distance measurement for AHC. As mentioned in Section I, GLR is widely used as such a measure to select the closest pair of clusters at every stage of AHC, but its sensitivity to data source variation in terms of accuracy results in the severe variability of the minimum possible level of DER across data sources. This can be confirmed in Figs. 3-4, where the minimum possible levels of DERs severely vary across the data sources considered. A possible key factor contributing to this robustness issue was analyzed

[11] GLR was used as an inter-cluster distance measure for AHC in this experiment.

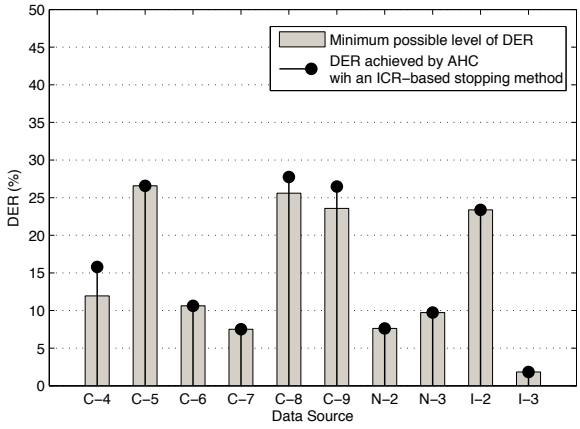


Figure 6.7: Comparison of the minimum possible levels of DERs for the evaluation data set (described in Section II) with the respective DERs obtained by AHC with the ICR-based stopping method with $\eta = 0.18603$. Average DER degradation by wrong estimation of the optimal stopping point is less than 1% (absolute) per data source.

in (226), where we found out that the large fraction of the segments shorter than 3 seconds in the input speech segments to AHC affected the minimum possible level of DER. To avoid such data dependency of the accuracy of the GLR-based inter-cluster distance measurement, we introduce here a simple modified version of AHC, namely selective AHC (SAHC).

SAHC first runs AHC (with the ICR-based stopping method) only on the segments longer than or equal to 3 seconds among the speech segments given for AHC, and then classifies the rest of the segments (shorter than 3 seconds) into one of the final clusters provided by the initial AHC, which is described in Algorithm 2. By doing this, the modified clustering strategy can enhance the accuracy of the GLR-based distance measurement during the initial AHC. Fig. 8 shows that AHC for a subset (of a given data source) containing only the segments longer than or equal to 3 seconds can, in general, achieve better performance than AHC for the entire given segments. Considering T_a in Table I in Section II, we can easily identify from the figure that such performance improvement is remarkable specifically for the data sources with many short segments, i.e., C-1, C-2, and I-1.

Fig. 9 shows SAHC performance for the evaluation data set. From the figure, we can see that SAHC is a reasonable strategy to tackle the robustness issue of the GLR-based inter-cluster distance measurement. The severe variability of the minimum level of DER across data sources is mitigated to some degree by SAHC. This mitigation was obtained significantly for C-8, C-9, and I-2. The overall DER improvement achieved by SAHC is 21.92% (relative) compared to simple AHC with the ICR-based stopping method.

6.5 Conclusions

In this chapter, we addressed the robustness issues of AHC to data source variation within the framework of speaker diarization, which are faced by the BIC-based stopping method and the GLR-based inter-cluster distance measurement in AHC. To tackle the problem caused by the BIC-based stopping method we proposed a novel ICR-based alternative. Furthermore, we introduced

Algorithm 2 Selective Agglomerative Hierarchical Clustering (SAHC)

Require: $\{\mathbf{x}_i\}, i = 1, ..., \hat{n}$: speech segments
$\hat{C}_i, i = 1, ..., \hat{n}', \hat{n}' \leq \hat{n}$: initial clusters
Ensure: $C_i, i = 1, ..., n$: finally remaining clusters
permutate $\{\mathbf{x}_i\}$ in the descending order of length
$\hat{C}_j \leftarrow \{\mathbf{x}_i\}$ such that $\{\mathbf{x}_i\}$ is a long speech segment ≥ 3 sec., $i = 1, ..., \hat{n}$ and $j = 1, ..., \hat{n}'$
$m = \hat{n}'$
do
  $i, j \leftarrow \arg\min \mathrm{GLR}(\hat{C}_k, \hat{C}_l), k, l = 1, ..., m, k \neq l$
  merge $\hat{C}_i$ to $\hat{C}_j$
  $m \leftarrow m - 1$
until DER reaches the lowest level
return $C_i, i = 1, ..., n$
$m = \hat{n}' + 1$
do
  $\hat{C} \leftarrow \{\mathbf{x}_m\}$
  $i \leftarrow \arg\min P(\hat{C}|\hat{C}_k), k = 1, ..., n$
  merge $\hat{C}$ to $\hat{C}_i$
  $m \leftarrow m + 1$
until $m > \hat{n}$
return $C_i, i = 1, ..., n$

Table 6.5: Global comparison (averaged DER) of AHC with the BIC-based stopping method, AHC with the ICR-based stopping method, and SAHC for the evaluation data set.

AHC (BIC)	AHC (ICR)	SAHC
24.49%	15.73%	12.28%

SAHC as a simple solution to tackle the severe variability of the minimum possible level of DER across data sources due to the sensitivity of the accuracy of the GLR-based inter-cluster distance measurement to data source variation. Through experimental results on excerpts obtained from meeting corpora, AHC with the ICR-based stopping method and SAHC were shown to outperform and be more robust to data source variation than basic AHC with the BIC-based stopping method. Table V presents performance comparison results of AHC with the BIC-based stopping method, AHC with the ICR-based stopping method, and SAHC for the evaluation data set. A reason for the improvements achieved by our proposed methods in terms of averaged DER across the data sources in the evaluation data set is because of the undesirable tendency of GLR where it tends to get larger as the total number of feature vectors within a pair of clusters under consideration increases was removed (in the case of AHC with the ICR-based stopping method), and the negative effect of the segments shorter than 3 seconds in the speech segments given for AHC on the minimum possible level of DER was mitigated (in the case of SAHC).

One potential future direction is to identify the lower bound for cluster size that guarantees ICR to be reliable as a statistical distance measure, more specifically as a homogeneity decision measure, between the clusters considered. In this chapter, we avoided the possibility that ICR

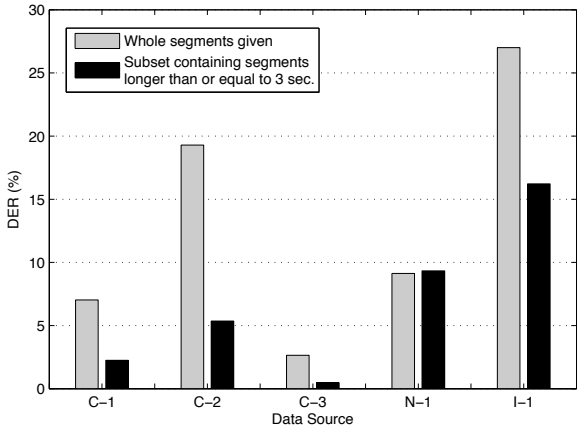


Figure 6.8: Minimum levels of DERs possibly achieved by AHC for the development data set. Comparison of performance for the whole speech segments given for AHC with that for a subset containing the segments longer than or equal to 3 seconds.

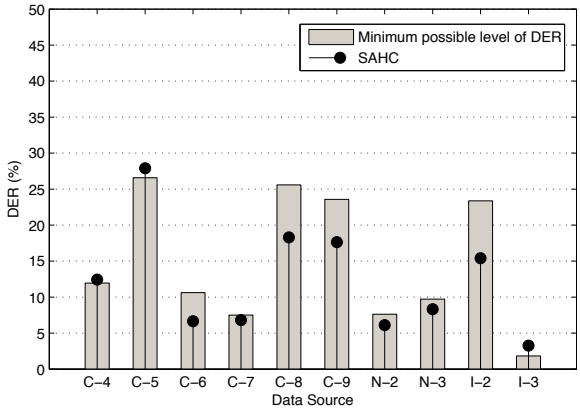


Figure 6.9: Comparison of the minimum possible levels of DERs for the evaluation data set with the respective DERs obtained by SAHC.

would not work properly, by checking ICR-based inter-cluster homogeneity starting from the pair of clusters merged at the last stage of AHC under the assumption that clusters at the later stages of AHC would be large enough for reliable ICR. This assumption worked for the meeting conversation excerpts used for the experiments presented in the chapter because most of the speakers involved in the conversations uttered longer than at least 30 seconds, which is empirically known to be long enough to represent speaker characteristics adequately. The assumption could be however broken for other data sources which have a preponderance of short speech segments that are inadequate to reveal the speaker characteristics completely.

Another future direction would be to search for the factors in a given data source for AHC that affect the reliability of the GLR-based inter-cluster distance measurement, other than the portion of short speech segments that we previously discovered. These could include the ratio of male and female speakers, the degree of intrinsic discernibility between speakers in terms of MFCC, and so on.

In this chapter, we assumed perfect speech/non-speech detection and speaker change detection. This is an appropriate assumption under the click-to-talk scenario. For future implementations in the SpeechLinks setting we need to allow for this complication.

Chapter 7

Prosody

Automatic Prosodic Event Detection using Acoustic, Lexical, and Syntactic Evidence

Spoken utterances are characterized not only by segment-level (spectral) correlates of each sound unit, but also by a variety of supra-segmental effects that operate at a level higher than the local phonetic context (109). The most prominent among these are:

- modulation of intensity to impart emphasis to certain syllables or words
- modulation of intonation patterns which reflect the class of the utterance (question, affirmation, etc.) as well as the speaker's intent and emotional state, and
- timing, which refers to subtle variations in the rate and length of syllables, coupled with pauses that serve to separate linguistic "phrases" within the utterances.

These supra-segmental effects occur at the syllable, word, and utterance level. Together, they encode rhythm, intonation, and lexical stress, which constitute the prosody of spoken utterances. As human listeners make heavy use of the above cues in the understanding process, they evidently carry a lot of information that is likely to be useful for spoken language understanding and generation systems (110) (111).

7.0.1 Motivation for prosodic event annotation

As mentioned above, prosody can be very useful because it encodes aspects of higher-level information not completely revealed by segmental acoustics. Below we list sample scenarios where prosody can play an important role in augmenting the abilities of spoken language systems.

1. *Speech act detection:* intonation patterns at the end of an utterance can provide an indication of specific speech acts or utterance categories (question, statement, exclamation, etc.).
2. *Word disambiguation:* knowledge of syllable stress or accent patterns can help in word or word-category disambiguation; a common example for this is the word *content*, which functions as a noun when stress is imparted to the first syllable (*con*-tent), and as an adjective when stress is imparted to the second syllable (con-*tent*).
3. *Speech recognition:* the correlation between accent / prominence patterns and words can be exploited to build joint lexical-prosodic models which can improve speech recognition performance in terms of reducing word-error rate (WER).

4. *Natural speech synthesis:* one of the challenges in natural sounding speech synthesis systems is to generate human-like prosody to accompany the segmental acoustic properties. This includes local effects (such as syllable accent), suitably timed boundaries, which reflect the syntactic structure of the utterance, as well as modulation of pitch at a global level to produce appropriate intonation patterns.

However, most current systems either completely disregard such information, or use it in limited, unprincipled ways for the simple reason that there is no established way to employ them. The main issues with using prosodic cues for spoken language applications are (a) the asynchronous nature of acoustic-prosodic features and consequently (b) the difficulty in modeling the relationship between the acoustic-prosodic features, segmental acoustics, lexical items and syntactic structure of the utterance. Having a symbolic representation of prosodic events in terms of discrete labels greatly simplifies the task of learning these relationships; however, such discretization, if not performed carefully, may result in loss of information from the prosodic tier.

The Tones and Break Indices (ToBI) annotation standard (112) (113) was developed in the early 1990s in an attempt to solve this problem, and to address the broader issue of representing prosodic events in spoken language in an unambiguous fashion. As such, ToBI is not a perfect scheme, and has been accused over the years of harboring several deficiencies (114), but it is the closest there is to a standard annotation system, and has been accepted as such by speech technologists and linguists working in this area.

7.0.2 The ToBI annotation scheme

The ToBI standard uses four inter-related "tiers" of annotation in order to capture prosodic events in spoken utterances:

1. The orthographic tier contains a plain-text transcription of the spoken utterance.

2. The tone tier marks the presence of *pitch accents* and *prosodic phrase boundaries*, which are defined as follows. A pitch accent can be broadly thought of as a prominence or stress mark. Two basic types of accents, high (H) and low (L) are defined, based on the value of the fundamental frequency (F0) with respect to its vicinity; more fine-grained accent marks, such as low-high (L+H*) and high-low (H+L*) are based on the shape of the F0 contour in the immediate vicinity of the accent. Prosodic phrase boundaries serve to group together semantic units in the utterance. These are divided in two coarse categories, weak *intermediate phrase boundaries* and *full intonational phrase boundaries*, each of which can be high (H) or low (L).

3. The break-index tier marks the perceived degree of separation between lexical items (words) in the utterance. Break indices range in value from 0 through 6, with 0 indicating no separation, or *cliticization*, and 6 indicating a full pause, such as at a sentence boundary. This tier is strongly correlated with phrase boundary markings on the tone tier - boundary locations usually score 4 or above on the break index tier.

4. The miscellaneous tier is used to annotate any other information relevant to the utterance that is not covered by the other tiers. This may include annotation of non-speech events such as disfluencies, etc.

A ToBI labeling guide along with several sample utterances annotated with ToBI labels is available in (113).

Although ToBI is by far the most well-known and widely used prosody annotation standard, it is not the only one in existence. INTSINT (115) (International Transcription System for Intonation) is a standard that is intended to function as an International Phonetic Alphabet (IPA) for describing the intonation contour of an utterance. Eight discrete symbols (T: top, M: mid, B: bottom, H: higher, L: lower, S: same, U: upstep, and D: downstepped) are used to parameterize the intonation contour. Of these, the first three (T, M, B) are *absolute*, i.e. defined with respect to the speaker's pitch range, while the other five (H, L, S, U, and D) are relative to the preceding target. The primary utility of INTSINT is to provide a parameterization of the overall intonation structure of the utterance, whereas ToBI is geared towards annotation of events that are linguistic in nature. As a result, INTSINT is more or less language independent, whereas a different version of ToBI has to be provided for each language (English, German, Japanese, Korean, and Greek are some of the languages for which complete ToBI system descriptions exist (116)).

Other prosody annotation systems include IViE (117) (Intonational Variation in English), which is derived from ToBI and is geared towards analysis and comparison of the intonational variation among different dialects / varieties of English, and TILT (118), which provides a numerical (continuous) parameterization of the intonation contour (as opposed to symbolic parameterization in INTSINT).

Although some annotation or parameterization systems may be better suited to specific tasks than ToBI (for instance, INTSINT and TILT are more suitable for parameterizing the intonation contour), ToBI is more general-purpose and is well suited for capturing the connection between intonation and prosodic structure. Another factor that has motivated most previous work in automatic prosodic annotation to use ToBI or its subsets is the wide availability of the Boston University Radio News Corpus, described in Section 7.1. This corpus has been hand-annotated with ToBI labels, and is now a standard data set for training and evaluating automatic prosody annotation techniques.

In this chapter, we focus on detecting a simpler subset of elements in the ToBI tone tier - specifically, we are interested in determining the presence or absence of pitch accents and phrase boundaries, regardless of their fine categories. As the title of the chapter suggests, these events can be detected not only from their acoustic correlates (energy, syllable duration, F0 range and contour, etc.), but also from the lexical and syntactic elements contained in an enriched textual representation of the utterance. Such a representation might include the orthography, part-of-speech (POS) and even a syntactic parse of the orthography of the utterance. The relationship of prosody to the acoustic, lexical, and syntactic structure of spoken utterances is discussed in further detail in following sections.

7.0.3 Previous work on ToBI-like prosodic event detection

Initial attempts at automatic detection of prosodic events are presented in the work by Wightman et al. (119) and Ross et al. (120). In (119), binary prominence and boundary labels were assigned to syllables based on posterior probabilities computed from acoustic evidence (such as F0, energy, and duration features) using a decision tree, combined with a probabilistic (bigram) model of accent and boundary patterns. Their method achieved an accuracy of 84% for prominence detection and 71% accuracy for boundary detection at the syllable level on the Boston University corpus. Thus, for prominence detection, they obtain performance levels that approach levels of agreement between human labelers (quoted as 86-94%) for this task. However, their boundary detection performance is lower than agreement levels between human annotators (95-98% for intonational phrase boundaries). In addition to prominence and boundary detection, they also conduct experiments on break index labeling, achieving an accuracy rate of 60% for exact index match and 90% for a

match within +/-1 of the true index.

In (120), the authors present an automatic pitch accent and boundary tone labeling system which predicts pitch accent labels and boundary tone types using a multi-level hierarchical model based on a decision tree framework. In addition to detecting presence vs. absence of pitch accents, they also attempt to perform fine-grained labeling of accent and boundary types. The fine pitch accent categories include high, downstepped, and low; fine boundary categories include L-L%, H-L%, and L-H%. With a single speaker training and test set, they obtained 87.7% accuracy for binary presence vs. absence of pitch accent at the syllable level. Pitch accents detected using the syllable level decision tree were then classified into fine categories using a pitch accent type classifier. They obtain an accuracy of 72.4% for the 3-class pitch accent categorization task, measured over the subset of syllables that were correctly marked by the pitch accent detector as being accented. However, since very few syllables carry the "low" pitch accent, this 3-way classifier was only marginally better than a chance-level accent type assignment that assigned a "high" pitch accent to all syllables (chance level accuracy was 71.8%). Their boundary tone classifer operates at the intonational phrase level. Intonational phrases are identified as those segments marked with a break index value of 4 or above on the ToBI break index tier. For boundary locations identified in this deterministic fashion, the 3-way boundary tone classifier produced boundary labels that were 66.9% accurate, as opposed to the chance level of 61.1%, where all boundary tones were labeled as L-L%. These accuracy figures are quoted at the intontational phrase level rather than at the word or syllable level.

Syrdal et al. (121) attempt to predict binary pitch accents and intonational boundary tones labels directly from lexical cues (text, punctuation, part-of-speech, etc.) using a text-to-speech (TTS) engine to obtain a "default" starting point for manual labelers. They determined that manual labeling of ToBI labels with this starting point was significantly faster than starting from "scratch" i.e. with no prior knowledge of pitch accent and boundary tone placement.

More recent efforts are reported in Chen et al. (122) and Ananthakrishnan et al. (123) The former used a GMM-based acoustic-prosodic model and an ANN-based syntactic-prosodic model built from POS tags in a maximum-likelihood framework to achieve binary pitch accent detection accuracy of 84.21% and intonational boundary detection accuracy of 93.07% at the word level. The latter experimented with an ASR-like structure for prosodic event detection, using a coupled-HMM structure to model the dynamic prosodic features and an n-gram based syntactic-prosodic model to obtain 75% agreement on the prominence detection task and 88% agreement on the boundary detection task (combining both intermediate and intonational phrase boundaries) at the syllable level. This system also has binary pitch accent and boundary event targets.

7.0.4 Our current approach

In this chapter, we attempt to combine different sources of information to improve accent and boundary detection performance. While our focus in this chapter is automatic annotation of corpora with prosodic event tags, we develop our model structure in a way that makes it easy to integrate with existing ASR architectures. We assume that, in addition to the speech data, we also have available the corresponding orthographic transcription annotated with POS tags. We collapse all categories of ToBI-style accent and boundary labels to single "accent" and "boundary" categories, respectively. Thus, we have two binary classification problems that we treat independently - presence vs. absence of pitch accents, and presence vs. absence of boundaries. We associate prosodic events with specific syllables, because the latter are traditionally regarded as the smallest linguistic units at which these phenomena manifest themselves.

Using statistics, we analyze the effect of these prosodic events on their acoustic correlates, such

as F0, short-time energy and timing cues; we also study their relationship to the syntactic part-of-speech, and to the lexical entities with which they correspond. Armed with this knowledge, we build classifiers that assign prosodic events to syllables from the unlabeled test data set using acoustic evidence extracted from the speech data. The classifiers also generate posterior probability scores for each class given the acoustic evidence. The relationship of prosody to syntactic POS and individual lexical items is exploited by building factored n-gram language models that capture such dependencies. Finally, for the pitch accent detection problem, we also incorporate prior knowledge from existing lexica that provide canonical pronunciation information (including stress marks) for a large body of words. Our work differs from previous efforts in the following respects:

- We use acoustic, lexical, and syntactic features as opposed to (119), who use only acoustic evidence, and (120; 121), who use only lexical and syntactic features. Our lexical and syntactic feature set is much simpler than that used in (122). In particular, we do not use syntactic phrase boundaries obtained from parsing the text for the boundary detection task.

- We detect pitch accent at the linguistic syllable level, similar to (119; 120), but different from (122), who do so at the word level. This is because prosodic events such as pitch accents are associated with specific syllables, rendering this approach more suitable for tasks such as word disambiguation, where two words may have the same phonetic pronunciation, but different syllable accent patterns (see example in Section 7.0.1)

- Previous work on boundary detection emphasizes intonational phrase boundaries only. Our boundary detection task is different, because we consider intermediate as well as intonational phrase boundaries as part of our "boundary" category. This is a much more difficult task than just intonational phrase boundary detection described in previous work.

- We use a maximum *a-posteriori* (MAP) framework for prosodic event detection as opposed to the maximum likelihood (ML) framework used in (122). Moreover, we use an n-gram structure for our prosodic language model, which makes for easier processing and decoding using the Viterbi algorithm, as well as integration with existing automatic speech recognition (ASR) systems. Another novelty of our work is the use of factored backoff to estimate smooth probabilities for the prosodic language model (see Section 7.3.2)

- The work described in (122) makes use of a prosodic lexicon that encodes all possible combinations of pitch accent and phrase boundaries for a given word. While this improves performance by restricting the search space, building such a lexicon is a time-consuming task that does not scale to other corpora. Our solution is to incorporate canonical stress information from a public domain electronic pronunciation dictionary within the statistical classification framework. We show that this corpus-independent approach leads to significant gains in pitch accent detection accuracy over using the lexical tokens alone.

The remainder of this chapter is organized as follows. Section 7.1 discusses the data corpus used, and the acoustic, syntactic, and lexical features extracted from the data for training and testing. Section 7.2 presents analyses of the acoustic, syntactic, and lexical correlates of accent and boundary events. Section 7.3 explains the basic architecture of our prosodic event detection system, and the assumptions that underly the structure. Section 7.4 details the experiments we conducted and the prosody recognition results we obtained. Finally, Section 7.5 contains a brief discussion of some of the open problems in this area, the limitations of our current approach, and how it may be improved and applied to spoken language systems.

7.1 Data corpus and features

The Boston University Radio News Corpus (BU-RNC) (124) is a database of broadcast news style read speech that contains ToBI-style prosodic annotations for part of the data. The availability of these annotations have made it the corpus of choice for most experiments on prosodic event detection and labeling, including all those cited in Section 7.0.3. The database contains speech from three female (*f1a*, *f2b* and *f3a*) and four male speakers (*m1b*, *m2b*, *m3b* and *m4b*). Data labeled with ToBI-style labels is available for 6 speakers, namely *f1a*, *f2b*, *f3a*, *m1b*, *m2b* and *m3b*, which amounts to about 3 hours of speech. In addition to the raw speech and prosodic annotation, the BU-RNC also contains

- orthographic (text) transcription corresponding to each utterance
- word- and phone-level time-alignments from automatic forced-alignment of the transcription and
- POS tags corresponding to each token in the orthographic transcription.

In order to obtain time-alignments at the linguistic syllable level, we syllabify the orthographic transcriptions using a deterministic algorithm based on the rules of English phonology (125), and since the resultant syllables are simply vowel-centric collections of the underlying phone sequences, we are able to generate syllable-level time alignments from phone-level alignments, which are available in the corpus.

For our experiments, we pooled all utterances that were ToBI-transcribed and created five cross-validation training and test sets. We then pruned the test sets so that no story repetitions by the same speaker co-existed in the training and test partitions of a given cross-validation set. This resulted in a training set size of 37,047 syllables and a test set size of 7,343 syllables, averaged across the five cross-validation sets. The average syllable vocabulary (number of unique syllables) of the training sets was 2,850, while that of the test sets was 1,623. The average number of out-of-vocabulary syllables in the test sets was 250 (15.4% relative to the test vocabulary). Of the syllables in the training sets, an average of 12,705 (34.3%) carried pitch accents, while 6,307 (17.0%) were associated with boundary events (counting both intermediate and intonational phrase boundaries). Of the syllables in the test sets, an average of 2,560 (34.9%) carried pitch accents, and 1,304 (17.7%) were associated with boundary events. Thus, the training and test sets exhibit similar chance levels for pitch accent and boundary events.

With the enriched transcriptions available in this corpus, we are then able to extract a variety of acoustic, lexical, and syntactic features as described below.

7.1.1 Acoustic features

Prosody has a marked effect on supra-segmental features such as F0, energy and timing in the vicinity of the event. Accent and boundary events are marked by exaggerated movements of the F0 contour. Accented syllables show an increase in the local energy profile. Pre-boundary syllable lengthening is a subtle timing variation found in the vicinity of boundary events (126). Our acoustic features are derived from these cues, and are listed below.

- Features derived from F0 include within-syllable F0 range (*f0_range*), difference between maximum and average within-syllable F0 (*f0_maxavg_diff/*), difference between minimum and average within-syllable F0 (*f0_avgmin_diff*), and difference between within-syllable average and utterance average F0 (*f0_avgutt_diff*).

- Features derived from timing cues include normalized vowel nucleus duration for each syllable (*n_dur*) and pause duration after the word-final syllable (*p_dur*, for boundary detection only)

- Features derived from energy include within-syllable energy range (*e_range*), difference between maximum and average within-syllable energy (*e_maxavg_diff*), and difference between minimum and average energy within the syllable (*e_avgmin_diff*).

The use of differences rather than absolute values for F0- and energy-related features serves to normalize the data against variation between speakers, especially between males and females, but preserves the variations produced by prosody. We normalized the syllable nucleus (vowel) duration on a per vowel-type basis, such that for each vowel-type, the normalized duration feature is zero mean and unit variance. This serves to eliminate absolute duration differences due to vowel-instrinsic properties (for example, the high-front vowel *iy* is usually much longer than the neutral *schwa*), while preserving differences due to pitch accent or boundary events. The pause duration feature used for boundary detection was not normalized. In addition to the above features, which are extracted directly from the speech data and the F0 track, we also include the number of phonemes in a syllable as an additional dimension to the acoustic feature vector. The complete set of acoustic features used for pitch accent and boundary detection is shown in Table 7.1. Thus, our acoustic features are encoded as nine-dimensional vectors, one for each syllable. We do not consider acoustic dependencies across syllables.

7.1.2 Lexical and syntactic features

As we will demonstrate in Sections 7.2 and 7.4, prosodic events in an utterance can be accurately predicted from the lexical and syntactic content of the underlying orthography (127). For example, content words such as nouns, adjectives and verbs are much more likely to contain prominent syllables than function words, such as articles and determiners. Phrase boundaries, too, are more likely to follow content words than function words. Similarly, certain syllables occur much more frequently in content words than in function words and are more likely to be accented than syllables that appear mostly in function words.

We use individual syllable tokens as lexical features and POS tags as syntactic features. For the accent detection problem, we also include the canonical stress pattern for the word; this is obtained from a standard pronunciation dictionary that includes stress marks. These features are used to build probabilistic models of prosodic event sequences. These "prosodic language models" have a structure similar to the word-level n-grams used in speech recognition, and are used to constrain and refine hypotheses generated by classifiers that operate only on the acoustic evidence.

7.2 Statistical Analyses of Acoustic, Lexical and Syntactic features

Before implementing classification algorithms using the features described in Section 7.1, we analyze these features in order to determine which ones are important for classification, and to verify if indeed they are capable of discriminating between the prosodic categories that we wish to separate. For the acoustic features, the former is accomplished using a feature selection algorithm, and the latter, using statistical hypothesis tests. In the case of lexical and syntactic features, we collect frequency counts to establish what types of lexical items or POS tags correspond well with accent and boundary events. These tests are conducted on the entire labeled corpus. Details of these analyses are presented below.

Table 7.1: Acoustic features ranked by importance

Accent features	**Boundary features**
n_dur	*p_dur*
f0_avgmin_diff	*n_dur*
e_avgmin_diff	*f0_maxavg_diff*
e_range	*f0_range*
f0_range	*e_range*
f0_maxavg_diff	*f0_avgmin_diff*
n_phones	*e_maxavg_diff*
f0_avgutt_diff	*e_avgmin_diff*
e_maxavg_diff	*f0_avgutt_diff*

7.2.1 Analysis of acoustic features

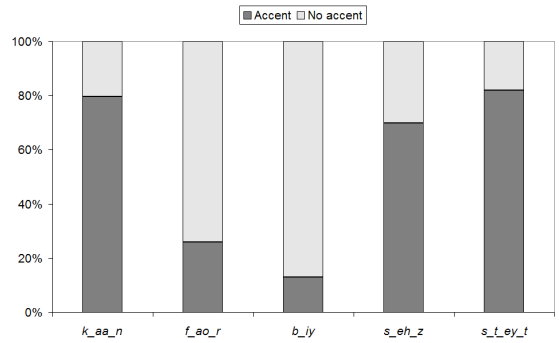
We conduct a feature selection experiment using the *information gain* criterion (128) in order to rank the acoustic features by importance. This was implemented using the WEKA machine learning toolkit (129). Table 7.1 lists acoustic features for pitch accent and boundary detection in decreasing order of importance based on this criterion. According to this ranking criterion, syllable nucleus duration is the most important determinant of pitch accent. Pause duration and nucleus duration are key indicators of boundary events. F0 and energy range also play an important role in discriminating between presence vs. non-presence of accent and boundary events.

Given the nature of our acoustic features, a simple two-way hypothesis test can also be conducted in order to determine whether the acoustic features are likely to be useful for classification. We test each feature independently in order to determine whether the mean value of the feature differs significantly across the positive and negative samples for each classification problem. Here, "positive" samples refer to syllables that carry a pitch accent or boundary; conversely, "negative" samples correspond to those syllables that do not carry accent or boundary events. We define the null and alternate hypotheses as follows.

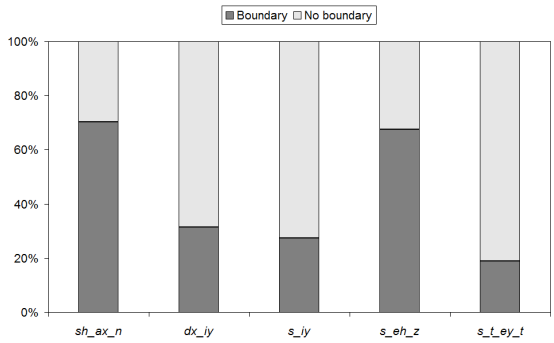
$H0$: the mean value of feature f_i does not differ between positive and negative samples

$H1$: the mean value of feature f_i differs between positive and negative samples

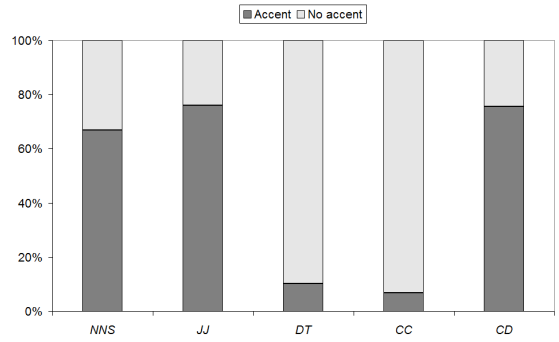
Analysis of variance (ANOVA) (130) is a commonly used statistical test to determine if two population means are different. however, standard ANOVA assumes that the variable being tested is normally distributed within each category label. In our case, most of the features are non-negative, hence this assumption is invalid. We therefore use a non-parametric form of ANOVA known as the Kruskal-Wallis test, which only makes the assumption that the samples are drawn independently. The significance level (p-value) reported by this test is lower than 0.001 for all but two acoustic features for pitch accent detection (*f0_avgutt_diff* and *e_maxavg_diff*), for which $p \leq 0.15$, corresponding to their low ranking by the information gain criterion. However, since they do carry some discrimination information, we include them in the acoustic feature set for pitch



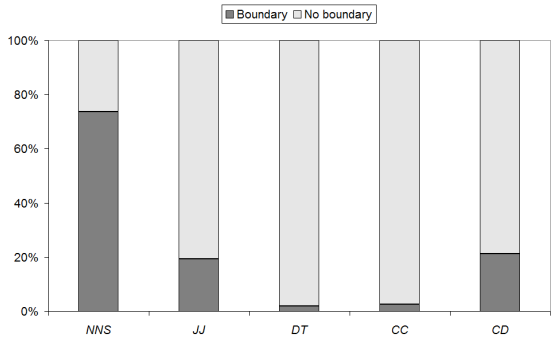
(a) Syllable-accent distribution



(b) Syllable-boundary distribution



(c) POS-accent distribution



(d) POS-boundary distribution

Figure 7.1: Unigram frequency distributions of selected syllable tokens and part-of-speech (POS) tags between positive and negative classes for pitch accent and boundary detection tasks. Figures show a clear preference of syllable tokens for specific categories. POS tags corresponding to content words (NNS, JJ, etc.) are much more likely to be associated with accented words than those that correspond to function words (DT, CC, etc.)

accent detection. The significance value reported by this test is below 0.001 for all features used in the boundary detection task. This indicates that the null hypothesis can be rejected with high confidence for most features. We conclude from this test that the acoustic features are likely to contain information that will discriminate between accent / boundary events and non-events.

7.2.2 Analysis of lexical and syntactic features

We use individual syllable tokens as lexical features and POS tags (at the word level) as syntactic features in order to predict prosodic events from non-acoustic evidence. We gather unigram frequency counts from these features in order to establish their relationship to accent and boundary events in the speech data. Figures 7.1(a) and 7.1(b) show the distribution of five randomly chosen syllable tokens between positive and negative samples of accent and boundary events, respectively. Each token appears more than 80 times in the training corpus. Figures 7.1(c) and 7.1(d) show a similar distribution for five randomly chosen POS tags in the corpus. Each of the tags considered appears several hundred, if not a few thousand times, in the corpus. In this case, since POS tags are associated with whole words rather than individual syllables, an accent label associated with a POS tag implies that one of the syllables that constitute the word is accented. Boundary events

are associated only with word-final syllables, hence a boundary label associated with a POS tag simply means that the final syllable of the corresponding word is at a boundary location.

These figures show a clear preference of certain syllable tokens and POS tags for specific prosodic events. For instance, in test data that statistically resemble the corpus used to compute the above statistics, there is approximately an 80% chance that the syllable token *k_aa_n* will be accented; on the other hand, the there is only a 13% chance that the token *b_iy* will be accented. Similarly, nouns (indicated by the tag *NNS*) have a 73% chance of being associated with boundary events, whereas adjectives (*JJ*) have only a 20% chance of being located at a prosodic phrase boundary. From this analysis of unigram frequency counts, we conclude that lexical and syntactic cues are likely to play an important role in recognition of accent and boundary events in speech.

7.3 Architecture of the prosodic event detector

Our prosodic event detector has a maximum *a-posteriori* (MAP) structure and is modeled on the lines of a standard automatic speech recognition (ASR) system. We seek the sequence of prosodic events that maximizes the posterior probability of the event sequence given the acoustic, lexical and syntactic evidence. In the following subsections, we develop the system architecture for each feature type separately, and then discuss feasible ways to merge them for performance improvement.

7.3.1 Prosodic event detection using acoustic evidence

We wish to find the sequence of prosodic events $\mathbf{P}^* = \{p_1^*, p_2^*, \ldots, p_n^*\}$ such that

$$\mathbf{P}^* = \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{P}|\mathbf{A}) \tag{7.1}$$

$$= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{A}|\mathbf{P})\, p(\mathbf{P}) \tag{7.2}$$

where $\mathbf{A} = \{a_1, a_2, \ldots, a_n\}$ is the sequence of acoustic feature vectors, one for each syllable. Since our acoustic-prosodic classifiers return posterior probabilities $p(p_i|a_i)$, we can classify each syllable independently, in which case we approximate Eq. 7.1 as

$$\mathbf{P}^* = \arg\max_{\mathbf{P}\in\mathcal{P}} \prod_{i=1}^{n} p(p_i|a_i) \tag{7.3}$$

We can incorporate context information by using the form of Eq. 7.2, where it is possible to model $p(\mathbf{P})$ as an n-gram of prosodic labels (we call this a *de-lexicalized* prosodic language model). For a trigram language model,

$$\begin{aligned}
\mathbf{P}^* = \arg\max_{\mathbf{P}\in\mathcal{P}} \quad & p(a_1|p_1)\, p(a_2|p_2)\, p(p_1)\, p(p_2|p_1) \\
& \cdot \prod_{i=3}^{n} p(a_i|p_i)\, p(p_i|p_{i-1}, p_{i-2}) \\
= \arg\max_{\mathbf{P}\in\mathcal{P}} \quad & \alpha(p_1|a_1)\, \alpha(p_2|a_2)\, p(p_1)\, p(p_2|p_1) \\
& \cdot \prod_{i=3}^{n} \alpha(p_i|a_i)\, p(p_i|p_{i-1}, p_{i-2}) \\
\text{where } \alpha(p_i|a_i) = & \frac{p(a_i|p_i)}{p(a_i)} = \frac{p(p_i|a_i)}{p(p_i)}
\end{aligned} \tag{7.4}$$

Eq. 7.4 is the architecture employed by Wightman et al. (119). They use a bigram prosodic language model with a decision tree providing the label posterior probabilities. In this chapter, we compare linear discriminant (LD), Gaussian mixture model (GMM) and neural network (NN) classifiers (131) (132) trained on acoustic features. Since the de-lexicalized prosodic LM has a binary vocabulary, it can be estimated very robustly even from small amounts of data. Thus, it is possible to model the prosody sequence using more context than it is to model word or syllable sequences; we use a 4-gram context for the prosodic LM. A small variation of this method is used for boundary detection; we specify that boundaries can only coincide with word-final syllables. Therefore, the terms $\alpha(p_i|a_i)$ are computed only for these syllables. For the word-initial and word-medial syllables, they are set to unimodal values so that the "no-boundary" event is always chosen.

7.3.2 Prosodic event detection using lexical evidence

The most likely sequence of prosodic events $\mathbf{P}^*$ given only the sequence of syllables $\mathbf{S}$ can be found as follows.

$$\begin{aligned} \mathbf{P}^* &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{P}|\mathbf{S}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{S},\mathbf{P}) \end{aligned} \tag{7.5}$$

where the joint distribution $p(\mathbf{S},\mathbf{P})$ can be modeled in an n-gram fashion; for example, a trigram approximation gives

$$\begin{aligned} p(\mathbf{S},\mathbf{P}) = p(s_1,p_1)\, p(s_2,p_2|s_1,p_1) \\ \cdot \prod_{i=3}^{n} p(s_i,p_i|s_{i-1},p_{i-1},s_{i-2},p_{i-2}) \end{aligned}$$

However, as detailed in Section 7.1, the vocabulary of syllables is quite large in relation to the training corpus, and it is difficult to robustly estimate this distribution even with the n-gram approximation. Moreover, the test data exhibits a significant out-of-vocabulary (OOV) rate for the syllables (15.4% relative to the test vocabulary). We therefore employ a factored backoff scheme (133), where the probability of the current syllable-event pair is conditioned on previous syllable-event pairs, but backs off to lower order distributions by dropping syllable tokens if reliable estimates cannot be obtained for the full conditional distribution. Since the syllable token sequence is known, the distribution $p(s_i,p_i|s_{i-1},p_{i-1},s_{i-2},p_{i-2})$ may be replaced, without loss of generality, by the expression $p(p_i|s_i,s_{i-1},p_{i-1},s_{i-2},p_{i-2})$. In this scheme, if an unseen factor, such as an OOV syllable, occurs as a conditioning variable in the term $p(p_i|s_i,s_{i-1},p_{i-1},s_{i-2},p_{i-2})$, the backed off estimate that does not contain this variable is substituted for the complete expression. Figure 7.2 shows the backoff graph we used for building the lexical-prosodic language model. The graph shows that we keep dropping lexical factors up to the point where we back off to the de-lexicalized prosodic LM described in Section 7.3.1. We use a fixed backoff path in this case. In practice, we use more (4-gram) history for the prosodic event factors and less (trigram) history for the syllable tokens.

7.3.3 Integrating information from a pronunciation lexicon

The CMU dictionary (134) is a widely available pronunciation lexicon of over 125,000 words that is commonly used in large-vocabulary ASR tasks. In addition to a phonetic transcription of each word, it also encodes the canonical stress pattern for each word. We can look up each word in

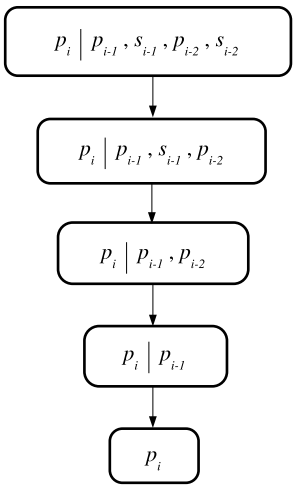


Figure 7.2: Backoff graph for estimating lexical-prosodic LM. At each step, we drop a conditioning variable. Lexical tokens are dropped first.

the test set from the lexicon and use the canonical stress pattern as another stream of evidence for pitch accent detection. We have, then, in addition to the syllable sequence $\mathbf{S}$, the sequence of canonical stress labels $\mathbf{L}$, whose elements are binary features. The problem then reduces to finding

$$\begin{aligned} \mathbf{P}^* &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{P}|\mathbf{S},\mathbf{L}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{S},\mathbf{L},\mathbf{P}) \end{aligned} \tag{7.6}$$

where the joint distribution can be approximated by its n-gram factors in a manner similar to that described in Section 7.3.2. The sparsity problem can again be alleviated by the use of factored backoff, where in this case there are three factors per syllable instead of two.

7.3.4 Prosodic event detection using syntactic evidence

We use syntactic evidence in the same way as we used lexical evidence to determine the most likely sequence of prosodic events.

$$\begin{aligned} \mathbf{P}^* &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{P}|\mathbf{POS}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{POS},\mathbf{P}) \end{aligned} \tag{7.7}$$

where, as above, the joint distribution can be expressed as a product of its n-gram factors

$$\begin{aligned} p(\mathbf{POS},\mathbf{P}) = {} & p(pos_1,p_1)\, p(pos_2,p_2|pos_1,p_1) \\ & \cdot \prod_{i=3}^{m} p(pos_i,p_i|pos_{i-1},p_{i-1},pos_{i-2},p_{i-2}) \end{aligned}$$

This syntactic-prosodic distribution is much easier to estimate than the lexical-prosodic distribution, because the vocabulary of POS tags is quite small (ca. 30-35 tags in all), and hence it is easy to obtain robust estimates even from limited amounts of training data.

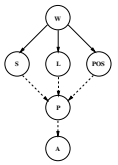


Figure 7.3: Directed graph illustrating dependencies among variables. **W** is the sequence of words; **S**, **L**, and **POS** the corresponding sequence of syllable tokens, canonical stress labels and part-of-speech tags, respectively; **P** the sequence of prosodic events and **A**, the sequence of acoustic-prosodic features. We treat the prosody labels as hidden variables influenced by (observed) lexical and syntactic features of the underlying orthography. The hidden prosodic event sequence generates acoustic observations.

One difference between the syntactic- and lexical-prosodic models is that the former is built at the word level. This is not an issue for determining boundaries, because the boundary is constrained to coincide with the final syllable of the word; hence, there can only be one boundary event per word. No such restriction is placed on pitch accents, as they can be associated with any syllable within the word. Thus, for the pitch accent detection task, the syntactic-prosodic model indicates whether some syllable within the word is accented, but does not provide information as to which one is. Even so, this model helps eliminate false-positive decisions by constraining all syllables within non-accented words to the "no-accent" tag. Note that we do not use information obtained from a syntactic parse of the orthography.

7.3.5 Combining acoustic, lexical and syntactic evidence

We would like to combine each of the above streams of evidence $\{\mathbf{A}, \mathbf{S}, \mathbf{POS}, \mathbf{L}\}$ in a principled fashion in order to maximize performance of the prosody recognizer. Figure 7.3 illustrates the dependencies between these variables in the form of a directed graph. The solid arrows indicate deterministic relationships, while the dotted ones represent probabilistic dependencies that we must model. We take the view that the word sequence $\mathbf{W}$, or equivalently, its features $\{\mathbf{S}, \mathbf{POS}, \mathbf{L}\}$ are responsible for generating the prosody of the spoken utterance, which in turn modulates the acoustic parameters such as F0, energy and duration, producing the acoustic feature sequence $\mathbf{A}$. In doing so, we ignore higher-level factors such as the utterance class (question, statement, etc.) and the speaker's emotional state that also play a role in determining the sequence of prosodic events. The observed variables in this graph are $\mathbf{W}$, the corresponding lexico-syntactic feature sequence $\{\mathbf{S}, \mathbf{POS}, \mathbf{L}\}$, and the acoustic feature sequence $\mathbf{A}$. The sequence of prosodic events $\mathbf{P}$ is to be

inferred. Hence,

$$\begin{aligned}\mathbf{P}^* &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{P}|\mathbf{A},\mathbf{S},\mathbf{L},\mathbf{POS}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{A},\mathbf{S},\mathbf{L},\mathbf{POS}|\mathbf{P})\, p(\mathbf{P}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{A}|\mathbf{P})\, p(\mathbf{S},\mathbf{L},\mathbf{POS}|\mathbf{P})\, p(\mathbf{P}) \qquad (7.8)\end{aligned}$$

where Eq. 7.8 follows from our assumption that the acoustic observations are conditionally independent of the lexical and syntactic features given the prosody labels. However, the distribution $p(\mathbf{S},\mathbf{L},\mathbf{POS}|\mathbf{P})$ cannot be robustly estimated because the joint vocabulary (ca. $2850\times2\times35$) is very large as compared to the available training data. We therefore use a naïve-Bayesian approximation such that the factors are easily and robustly estimated.

$$p(\mathbf{S},\mathbf{L},\mathbf{POS}|\mathbf{P}) \approx p(\mathbf{S},\mathbf{L}|\mathbf{P})\, p(\mathbf{POS}|\mathbf{P}) \qquad (7.9)$$

Note that the feature sequence $\mathbf{L}$ is not available for the boundary detection task. Making this approximation, and substituting Eq. 7.9 into Eq. 7.8 gives

$$\begin{aligned}\mathbf{P}^* &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{A}|\mathbf{P})\, p(\mathbf{S},\mathbf{L}|\mathbf{P})\, p(\mathbf{POS}|\mathbf{P})\, p(\mathbf{P}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} p(\mathbf{A}|\mathbf{P})\, p(\mathbf{S},\mathbf{L},\mathbf{P})\, p(\mathbf{POS}|\mathbf{P}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} \frac{p(\mathbf{A}|\mathbf{P})}{p(\mathbf{P})}\, p(\mathbf{S},\mathbf{L},\mathbf{P})\, p(\mathbf{POS},\mathbf{P}) \qquad (7.10) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} \frac{p(\mathbf{P}|\mathbf{A})}{p^2(\mathbf{P})}\, p(\mathbf{S},\mathbf{L},\mathbf{P})\, p(\mathbf{POS},\mathbf{P}) \\ &= \arg\max_{\mathbf{P}\in\mathcal{P}} \beta(\mathbf{P}|\mathbf{A})\, p(\mathbf{S},\mathbf{L},\mathbf{P})\, p(\mathbf{POS},\mathbf{P}) \\ &\qquad \text{where } \beta(\mathbf{P}|\mathbf{A}) = \frac{p(\mathbf{P}|\mathbf{A})}{p^2(\mathbf{P})}\end{aligned}$$

The combined recognition model hence reduces to a product of the individual acoustic, lexical and syntactic models, respectively.

7.4 Experimental results

We conduct a number of prosodic event detection experiments using acoustic, lexical and syntactic cues, as discussed in Section 7.3. In this section, we describe our experimental setup and recognition results for the individual and combined models. All performance figures in this section are obtained using 5-fold cross validation with training and test splits as described in Section 7.1. For each classification experiment, we list the accent and boundary detection accuracy as well as the corresponding false positive (FP) percentages. For the boundary detection task, we list overall detection accuracy as a fraction of all syllables, as well as word-final detection accuracy as a fraction of just the word-final (WF) syllables. The latter is a more useful metric, since word-initial and -medial syllables are always forced to the "no-boundary" category by our classifiers. We also report confidence intervals in terms of significance values (p-values) wherever we make comparisons between the performance of different classifiers and feature sets. We use the Wilcoxon signed rank test to compute significance values, because it is non-parametric, works with small sample sizes, and makes no assumptions regarding the distribution of the values (in this case, accuracy rates) being compared.

7.4.1 Baseline

We set up a simple baseline based on the chance level of pitch accent and boundary events computed from the training data. Approximately 34% of training syllables carry an accent, while only about 17% of syllables coincide with boundaries. We form a lattice where each test syllable can take on positive or negative labels with the corresponding *a-priori* chance level computed from the training corpus and rescore this lattice with the de-lexicalized prosodic LM to obtain a baseline system. The baseline pitch accent and boundary detection accuracies were 67.94% and 82.82% (overall), respectively. Note that our baseline boundary detection accuracy (based on the chance level for boundaries) is higher than the IPB detection accuracy of 71% reported in (119) for the radio news task. However, unlike (119), we provide figures for intermediate and intonational boundaries together.

7.4.2 Acoustic prosodic event detector

We employed three different classifiers (LD, GMM and NN) to obtain the prosody labels from the acoustic evidence. The GMM and NN classifiers also provide posterior probabilities for the prosodic events given the evidence. We first tested these classifiers in "independent-syllable" mode (Eq. 7.3), and chose the best performing one for combination with the de-lexicalized prosodic LM.

7.4.2.1 Linear discriminant classifier

The LD classifier was used to obtain a simple baseline for classification based on acoustic evidence. The weights are trained using standard batch least-squares (the "pseudoinverse" method). This classifier achieved an independent syllable classification accuracy of 71.15% for pitch accent detection and 89.30% (overall) for the boundary detection task.

7.4.2.2 Gaussian mixture-model classifier

We trained GMM-based classifiers for pitch accent and boundary events using the EM algorithm. The number of component mixtures was chosen using the Bayesian Information Criterion (BIC). Although it not optimal in the sense of minimizing classification error, the BIC score provides a convenient way to select the number of mixtures based on a minimum-description length criterion. Specifically, the BIC score is simply the log-likelihood of the training data given the GMM parameters penalized by a function of the number of parameters and training samples. Based on this metric, the best choice for the number of mixtures was 18. This classifier achieved an independent syllable classification accuracy of 72.18% for the pitch accent detection task and 89.41% (overall) for the boundary detection task.

7.4.2.3 Neural network classifier

The small difference in performance between the GMM and LD classifiers despite the large difference in the number of model parameters suggests that the acoustic features are not modeled well by GMMs. This led us to use a neural network for classifying prosodic features. We built a two-layer feedforward neural network with 9 input units, 25 hidden units, and 2 output units, one for each class. We used linear activation for the input units, sigmoidal activation for the hidden units and softmax activation for the output units. The neural network was trained using standard

Table 7.2: Acoustic prosody recognizer: performance

	Accent	Accent FP
LD	71.15%	7.24%
GMM	72.18%	9.75%
NN	74.10%	8.64%
NN + de-lex LM	80.07%	10.14%

	Boundary		Boundary FP	
	All	**WF**	**All**	**WF**
LD	89.30%	83.47%	1.02%	1.68%
GMM	89.41%	83.65%	2.20%	3.63%
NN	89.99%	84.61%	2.30%	3.80%
NN + de-lex LM	89.59%	83.95%	5.09%	8.41%

backpropagation. This classifier achieved an independent syllable classification accuracy of 74.10% for pitch accent detection task and 89.99% for the boundary detection task.

7.4.2.4 Acoustic classifier + de-lexicalized LM

We combine posterior label probabilities from the best performing acoustic classifier, the neural network, with label sequence constraints imposed by a 4-gram de-lexicalized prosodic LM. This is achieved by constructing a sausage lattice with prosodic variants of each syllable forming the lattice arcs. Each arc is weighted by the posterior probability assigned by the acoustic classifier (neural network). This lattice is then rescored with the n-gram de-lexicalized prosodic LM. This resulted in an absolute accuracy improvement of 5.97% for pitch accent detection (significant at $p \leq 0.05$). However, accuracy actually decreased by 0.4% ($p \leq 0.05$) for the boundary detection task. This is probably due to the fact that boundary events are quite far apart, and their context cannot be captured by narrow n-gram models. In the BU corpus, boundary events occur on average once every 6-7 syllables; constructing such long range n-gram LMs is not feasible even for a binary vocabulary. We found this to be the case empirically as well; a 5-gram LM performed worse than the 4-gram with which we report the above results. Table 7.2 summarizes classification accuracy results using acoustic evidence.

7.4.3 Lexical prosodic event detector

In this setup, we attempt to uncover prosodic events using only lexical evidence, i.e. the syllable tokens and, for the accent detection task, the canonical stress sequence obtained from a pronunciation lexicon. The lexical-prosodic LMs were implemented using a factored backoff scheme according to Figure 7.2 in order to alleviate problems due to data sparsity. We built these models using the *fngram* tools that are part of the well-known SRILM toolkit (135). The test transcriptions were used to construct unweighted lattices for each utterance; these lattices have a sausage structure and encode all possible combinations of syllable tokens and prosodic events for the corresponding

Table 7.3: Lexical/Syntactic prosody recognizer: performance

	Accent	**Accent FP**
Tokens only	82.92%	8.09%
Incl. lexicon	85.17%	8.65%
Syntax only	70.70%	2.13%

	Boundary		**Boundary FP**	
	All	**WF**	**All**	**WF**
Tokens only	85.73%	77.59%	4.08%	6.75%
Syntax only	87.99%	81.31%	2.28%	3.77%

utterances. They were then scored with the language model and the best paths through the lattices were obtained using Viterbi search. This yielded the most likely sequence of prosodic events.

This experiment was conducted both with and without the canonical stress patterns from the pronunciation lexicon (for pitch accent detection) in order to study the effects of such *a-priori* knowledge on system performance. The results are summarized in Table 7.3. We observe that classification accuracy from syllable tokens alone exceeds the performance of a purely acoustic evidence based classifier by a significant margin (82.92% vs. 80.07%, $p \leq 0.05$). However, prediction of boundary events using lexical evidence alone was 4.26% less accurate ($p \leq 0.05$) than predicting them using acoustic evidence. We also note, for the accent classification task, that the use of a pronunciation lexicon leads to an absolute classification accuracy improvement of 2.25% ($p \leq 0.05$) over a classifier that uses only the syllable tokens.

7.4.4 Syntactic prosodic event detector

The structure of the syntactic prosodic event detector is similar to that of the lexical prosody recognizer, except for two differences. The first is that we use a standard backoff trigram to model the joint distribution of POS tags and prosodic events, which are treated as compound tokens. As mentioned earlier, the POS vocabulary is quite small and no sparsity issues are likely to arise even with a relatively small training set. The second difference is that this recognizer detects prosodic events at the word level rather than at the syllable level. This is not an issue for the boundary detection task, as we force all non-word-final syllables to the negative label. However, the syntactic-prosodic LM does not influence classification of individual syllables that comprise the accented variant of a word. Syllables within an accented word are assigned pitch accents according to the chance level observed in training data. Table 7.3 summarizes accent and boundary detection accuracy for this recognizer. As expected, we observe that this method results in a performance gain over the lexical classifier for the boundary classification task (87.99% vs. 85.73%, $p \leq 0.05$), but produces significantly worse results on the pitch accent detection task (70.70% vs. 85.17%, $p \leq 0.05$), only slightly better than the baseline. This is expected, because for multisyllabic words that are identified as being accented, the syntactic model does not predict which syllable carries the pitch accent.

Table 7.4: Combined prosody recognizer: performance

	Accent	Accent FP
Baseline	67.94%	11.33%
Acoustic + Lexical (with pron.)	86.37%	7.64%
Acoustic + Syntactic	76.04%	7.25%
Acous. + Lex. + Syn. (no pron.)	86.06%	6.58%
Acous. + Lex. + Syn. (with pron.)	86.75%	8.08%
Word-level baseline	72.73%	6.66%
Combined system (word-level)	84.59%	9.33%

	Boundary		Boundary FP	
	All	**WF**	**All**	**WF**
Baseline	82.82%	72.78%	0.17%	0.28%
Acoustic + Lexical	90.41%	85.31%	4.63%	7.65%
Acoustic + Syntactic	91.61%	87.29%	4.61%	7.61%
Acous. + Lex. + Syn.	91.38%	86.91%	5.51%	9.11%

7.4.5 Combined acoustic, lexical and syntactic prosodic event detector

Having tested prosodic event detection performance with each feature stream separately, we now combine them in accordance with Eq. 7.10. The issue of combining the syntactic-prosodic LM and lexical-prosodic LM arises again, because the former is built at the word level and the latter, at the syllable level. We address this problem by representing the syntactic lattice as a finite-state acceptor (FSA) and the word-to-syllable mapping as a finite-state transducer (FST). Scores from the acoustic model are embedded in the FST. The syntactic FSA is scored with the syntactic-prosodic LM and then composed with the mapping FST. This produces an syllable-token level FSA that incorporates syntactic and acoustic scores, which is finally rescored with the lexical-prosodic LM to obtain the best sequence of labels. We implemented the composition and other FSM operations with the AT&T FSM toolkit (136).

In addition to combining all feature streams, we also tested classifiers that used only acoustic and lexical features, and another that combined only acoustic and syntactic features. These experiments were conducted in order to examine the effects of the assumption underlying Eq. 7.9. Table 7.4 summarizes the performance of the combined feature classifiers. We note that the combining all feature streams produces the most accurate pitch accent classification results, whereas boundary classification accuracy is highest for the classifier that combines only acoustic and syntactic evidence. Addition of the lexical feature stream actually decreases performance by 0.23%; however, this result was not significant at $p \leq 0.05$. This lack of performance improvement probably arises from the fact that lexical features (syllable tokens) are poorer indicators of boundary events than POS tags and therefore do not provide any additional information over the syntactic features. We note that for pitch accent detection, the combined system that uses canonical stress patterns from the pronunciation dictionary performs better than the combined system that does not use these stress patterns (86.75% vs. 86.06%, $p \leq 0.05$). Finally, we also derive word-level pitch

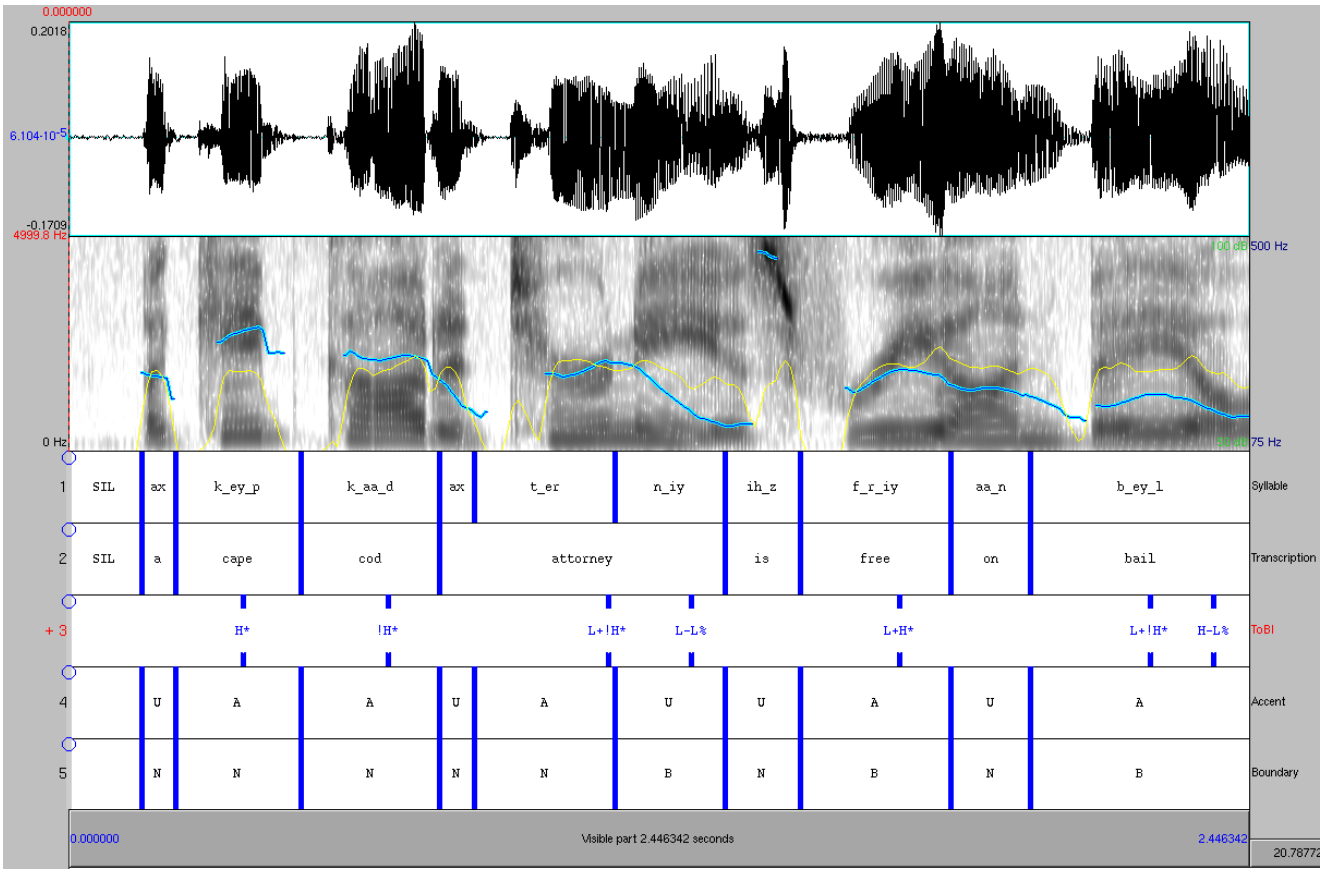


Figure 7.4: Sample prosodic event detector output for utterance *f1as01p1*. The first 2.4 seconds of the utterance are shown. Tier 1 shows the speech signal; tier 2 shows the spectrogram with superimposed F0 and intensity tracks; tier 3 shows syllable-level transcription with time-alignments; tier 4 shows time-aligned word-level transcriptions; tier 5 shows ToBI pitch accents and boundaries as annotated in the corpus; tier 6 shows accent events assigned to syllables (U: unaccented, A: accented); tier 7 shows boundary events aligned with syllables (N: no boundary, B: boundary)

accent detection performance from syllable level annotations - a word carries a pitch accent if any syllable within that word is identified as carrying an accent. The baseline performance for this task was 72.73%, and on combining acoustic, lexical and syntactic features, we obtained a significant performance improvement up to 84.59%.

Figure 7.4 shows a sample output of the combined prosodic event detector for the first 2.4 seconds of the utterance *f1bs01p1* in a Praat TextGrid display. The figure contains 7 tiers. Tier 1 shows the speech signal; tier 2 shows the spectrogram with superimposed F0 and intensity tracks; tier 3 shows syllable-level transcription with time-alignments; tier 4 shows time-aligned word-level transcriptions; tier 5 shows ToBI-style pitch accents and boundaries as annotated in the corpus; tier 6 shows accent events assigned to syllables (U = unaccented, A = accented); tier 7 shows boundary events aligned with syllables (N = no boundary, B = boundary). In this example, all pitch accent events were correctly identified and assigned to their corresponding syllables, but there is one error in the boundary tier, where the syllable *f_r_iy* has been assigned a boundary event, where, in fact, there is none (this may be attributed to the statistical nature of the boundary detector, similar to

word errors in ASR - our system is, after all, a "speech recognizer" for prosodic events).

7.5 Discussion and future work

In this chapter, we developed a pitch accent detection system that obtained an accuracy of 86.75%, and a prosodic phrase boundary detector that obtained an accuracy of 91.61% at the linguistic syllable level on a human-annotated test set derived from the BU-RNC. Both systems approach the agreement level between human labelers for these tasks. We incorporated acoustic, lexical and shallow syntactic features within a MAP framework, which, combined with the n-gram prosodic language models, makes it easy to integrate with existing ASR systems. We determined from our experiments that lexical syllable tokens are useful for pitch accent detection, but not so effective for boundary detection. On the other hand, syntactic POS tags play an important role in boundary detection, but are not as useful for predicting pitch accent events. We also determined that canonical stress labels from a pronunciation dictionary are useful for pitch accent detection.

Our pitch accent detector performs better than that described in (119) (86.75% vs. 84% at the syllable level). Although (122) also report 84.21% accuracy on this task, the systems are not directly comparable, as we assign pitch accents to syllables, while their system operates at the word level. The work by Ross et al. (120) reports pitch accent detection accuracy of 87.7% at the syllable level; however, this is on a very small test set using data from only one speaker, whereas we report results on 6 speakers (3 male, 3 female) using speaker independent prosody models.

Our boundary detection task is more difficult than that described in previous work, because they focus only on intonational phrase boundary detection, whereas we consider both intermediate and intonational phrase boundaries as valid boundary events. Our boundary detection performance significantly exceeds that reported in (119) (91.61% vs. 71% at the syllable level), but lags that reported in (122) (87.29% vs. 93.07% at the word level). However, the figures quoted in (119; 122) are for intonational phrase boundary detection only. Also, unlike (122), we do not use phrase opening/closing information from a syntactic parse of the text for the boundary detection task. Our task cannot be compared with the boundary tone classification problem described in (120), because they perform classification of boundary tones that have been deterministically identified from the ToBI break index tier, and not boundary tone detection itself.

As discussed in Section 7.0.1, automatic recognition of pitch accent and boundary events in speech can be very useful for tasks such as word disambiguation, where a group of words may be phonetically similar but differ in placement of pitch accent, or in their location with respect to a boundary. At a higher level, knowledge of these prosodic events can be useful for spoken language understanding systems. For instance, in building a speech-to-speech translation system, we would like the supra-segmental structure in the target language to be equivalent to that of the utterance in the source language. Mapping prosodic events to a finite set of categories is a good starting point for this task.

There are several open problems that still need to be addressed. First, we work with binarized versions of the ToBI label set, disregarding the fine categories i.e. types of pitch accents and boundaries. These fine categories are annotated on the basis of the intonation pattern in the vicinity of the syllable associated with the prosodic event. However, to distinguish between these types using automatically extracted features is a difficult problem because (a) we rely on syllable time alignments generated from automatic forced alignment of the speech, which is not very accurate and (b) intonation patterns used by human annotators to make these fine distinctions often occur in an asynchronous fashion, and do not always lie within the time-window indicated by forced alignment. As a result, extracting reliable features for distinguishing fine categories becomes very

difficult. Indeed, most previous work, including that cited in this chapter, focuses on binary pitch accent and boundary tone detection. An exception is (120), who report results on fine categorization of pitch accents and boundary tones. However, as their results show, fine categorization does not yield significant improvement over chance level category assignment (72.4% vs. 71.8%) for 3-way pitch accent categorization. For boundary tone locations deterministically identified from the ToBI break index tier, 3-way classification of tone category was somewhat better (66.9% vs. 61.1% for chance level assignment).

Second, for our current approach to be useful, we require a training corpus that is annotated with pitch accent and boundary labels. The BU-RNC is a broadcast news style corpus, and models trained with this data may not generalize well to spontaneous speech, which is usually the input for most spoken language understanding systems. We would therefore like to experiment with semi-supervised and unsupervised techniques to perform the labeling task where such annotations are not available in the training set. Previous work on unsupervised prosodic event detection has focused exclusively on acoustic evidence (137). In (138), we describe an unsupervised algorithm for accent and boundary event detection using acoustic, lexical and part-of-speech evidence. The algorithm described in that chapter uses information from an unsupervised clustering process to bootstrap lexical and syntactic probability models for improved performance.

Finally, in our current approach, we assume that the orthography and syntactic features (POS tags) corresponding to the spoken utterances are available. In many cases, however, we have only the speech utterance and wish to detect prosodic events directly from the acoustic signal, either for improving speech recognition performance, or to extract other paralinguistic information such as speech acts, emotion, etc. One possible approach is discussed in Hasegawa-Johnson et al. (139), who use a lexical-syntactic-prosodic LM in order to simultaneously obtain word hypotheses as well as accent and boundary labels. More generally, incorporating prosodic cues in ASR to improve word recognition performance is a difficult problem, and we would like to see if operating at a lower level of granularity (such as accent and boundary events) will improve performance. In recent experiments (140), we obtained modest but statistically significant word-error rate improvement by re-ranking ASR N-best lists with prosody models similar to the ones described in this chapter.

Chapter 8

Prosody in Maximum Entropy Framework

Exploiting Acoustic and Syntactic Features for Automatic Prosody Labeling in a Maximum Entropy Framework

Prosody is generally used to describe aspects of a spoken utterance's pronunciation which are not adequately explained by segmental acoustic correlates of sound units (phones). The prosodic information associated with a unit of speech, say, syllable, word, phrase or clause, influences all the segments of the unit in an utterance. In this sense they are also referred to as suprasegmentals (141) that transcend the properties of local phonetic context.

Prosody encoded in the form of intonation, rhythm and lexical stress patterns of spoken language, conveys linguistic and paralinguistic information such as emphasis, intent, attitude and emotion of a speaker. On the other hand, prosody is also used by speakers to provide cues to the listener and aid in the appropriate interpretation of their speech. This facilitates a method to convey the intent of the speaker through meaningful chunking or phrasing of the sentence, and is typically achieved by breaking long sentences into smaller prosodic phrases. Two key prosodic attributes described above include **prominence** and **phrasing** (142).

Prosody in spoken language correlates with acoustic and syntactic features. Acoustic correlates of duration, intensity and pitch, such as syllable nuclei duration, short time energy and fundamental frequency (f0) are some of the acoustic features that are used to express prosodic prominence or stress in English. Lexical and syntactic features such as parts-of-speech, syllable nuclei identity, syllable stress of neighboring words have also been shown to exhibit high degree of correlation with prominence. Humans realize phrasing acoustically by pausing after a major prosodic phrase, accentuating the final syllable in a phrase, and/or by lengthening the final syllable nuclei before a phrase boundary. Prosodic phrase breaks typically coincide with syntactic boundaries (143). However, prosodic phrase structure is not isomorphic to the syntactic structure (144; 145).

Incorporating prosodic information can be beneficial in speech applications such as text-to-speech synthesis, automatic speech recognition and natural language understanding, dialog act detection and even speech-to-speech translation. Accounting for the correct prosodic structure is essential in text-to-speech synthesis to produce natural sounding speech with appropriate pauses, intonation and duration. Speech understanding applications also benefit from being able to interpret the recognized utterance through the placement of correct prosodic phrasing and prominence. Speech-to-speech translation systems can also greatly benefit from the marking of prosodic phrase boundaries, for e.g., providing this information could directly help in building better phrase-based

statistical machine translation systems. The integration of prosody in these applications is preempted by two main requirements:

1. A suitable and appropriate representation of prosody (e.g., categorical or continuous)
2. Algorithms to automatically detect and seamlessly integrate the detected prosodic structure in speech applications

Prosody is highly dependent on the individual speaker style, gender, dialect and phonological factors. Non-uniform acoustic realizations of prosody are characterized by distinct intonation patterns and prosodic constituents. These distinct intonation patterns are typically represented using either, symbolic or parametric prosodic labeling schemes such as Tones and Break Indices (ToBI) (146), TILT intonational model (147), Fujisaki model (148), Intonational Variation in English (IViE) (149) and International Transcription System for Intonation (INTSINT) (150). These prosodic labeling approaches provide a common framework for characterizing prosody and hence facilitate development of algorithms and computational modeling frameworks for automatic detection and subsequent integration of prosody within various speech applications. While detailed categorical representations are suitable for text-to-speech synthesis, speech and natural language understanding tasks, simpler prosodic representations in terms of raw or speaker normalized acoustic correlates of prosody have also been shown to be beneficial in many speech applications such as disfluency detection (151), sentence boundary detection (152), parsing (153) and dialog act detection (154). As long as the acoustic correlates are reliably extracted under identical conditions during training and testing, an intermediate symbolic or parametric representation of prosody can be avoided, even though they may provide additional discriminative information if available. In this work, we use the ToBI labeling scheme for categorical representation of prosody.

Prior efforts in automatic prosody labeling have utilized a variety of machine learning techniques, such as decision trees (142; 155), rule-based systems (156), bagging and boosting on decision trees (157), hidden markov models (158), coupled HMMs (159), neural networks (160) and conditional random fields (161). These algorithms typically exploit lexical, syntactic and acoustic features in a supervised learning scenario to predict prosodic constituents characterized through one of the aforementioned prosodic representations.

The interplay between acoustic, syntactic and lexical features in characterizing prosodic events has been successfully exploited in text-to-speech synthesis (162; 163), dialog act modeling (164; 165), speech recognition (160) and speech understanding (142). The procedure in which the lexical, syntactic and acoustic features are integrated plays a vital role in the overall robustness of automatic prosody detection. While generative models using HMMs typically perform a front-end acoustic-prosodic recognition and integrate syntactic information through back-off language models (159; 160), stand-alone classifiers use a concatenated feature vector combining the three sources of information (161; 166). We believe that a discriminatively trained model that jointly exploits lexical, syntactic and acoustic information would be the best suited for the task of prosody labeling. We present a brief synopsis of the contribution of this chapter in the following section.

8.0.1 Contributions of this work

We present a discriminative classification framework using maximum entropy modeling for automatic prosody detection. The proposed classification framework is applied to both prominence and phrase structure prediction, two important prosodic attributes that convey vital suprasegmental

information beyond the orthographic transcription. The prominence and phrase structure prediction is carried out within the Tones and Breaks Indices (ToBI) framework designed for categorical prosody representation. We perform automatic pitch accent and boundary tone detection, and break index prediction, that characterize prominence and phrase structure, respectively, with the ToBI annotation scheme.

The primary motivation for the proposed work is to exploit lexical, syntactic and acoustic-prosodic features in a discriminative modeling framework for prosody modeling that can be easily integrated in a variety of speech applications. The following are some of the salient aspects of our work:

8.0.1.1 Syntactic features

- We propose the use of novel syntactic features for prosody labeling in the form of supertags which represent dependency analysis of an utterance and its predicate-argument structure, akin to a shallow syntactic parse. We demonstrate that inclusion of supertag features can further exploit the prosody-syntax relationship compared to that offered by using parts-of-speech tags alone.

8.0.1.2 Acoustic features

- We propose a novel representation scheme for the modeling of acoustic-prosodic features such as energy and pitch. We use n-gram features derived from the quantized continuous acoustic-prosodic sequence that is integrated in the maximum entropy classification scheme. Such an n-gram feature representation of the prosodic contour is similar to representing the acoustic-prosodic features with a piecewise linear fit as done in parametric approaches to modeling intonation.

8.0.1.3 Modeling

- We present a maximum entropy framework for prosody detection that jointly exploits lexical, syntactic and prosodic features. Maximum entropy modeling has been shown to be favorable for a variety of natural language processing tasks such as part-of-speech tagging, statistical machine translation, sentence chunking, etc. In this work we demonstrate the suitability of such a framework for automatic prosody detection. The proposed framework achieves state-of-the-art results in pitch accent, boundary tone and break index detection on the Boston University (BU) Radio News Corpus (167) and Boston Directions Corpus (BDC) (168), two publicly available read speech corpora with prosodic annotation.

- Our framework for modeling prosodic attributes using lexical, syntactic and acoustic information is at the word level, as opposed to syllable level. Thus, the proposed automatic prosody labeler can be readily integrated in speech recognition, text-to-speech synthesis, speech translation and dialog modeling applications.

The rest of the chapter is organized as follows. In section 8.1 we describe some of the standard prosodic labeling schemes for representation of prosody, particularly, the ToBI annotation scheme that we use in our experiments. We discuss related work in automatic prosody labeling in section 8.2 followed by a description of the proposed maximum entropy algorithm for prosody labeling in section 8.3. Section 8.4 describes the lexical, syntactic and acoustic-prosodic features used in our framework and section 8.5.1 describes the data used. We present results of pitch accent and

boundary tone detection, and break index detection in sections 8.6 and 8.7, respectively. We provide discussion of our results in section 8.8 and conclude in section 8.9 along with directions for future work.

8.1 Prosodic labeling standards

Automatic detection of prosodic prominence and phrasing requires appropriate representation schemes that can characterize prosody in a standardized manner and hence facilitate design of algorithms that can exploit lexical, syntactic and acoustic features in detecting the derived prosodic representation. Existing prosody annotation schemes range from those that seek comprehensive representations for capturing the various multiple facets of prosody to those that focus on exclusive categorization of certain prosodic events.

Prosodic labeling systems can be categorized into two main types: linguistic systems, such as ToBI (146), which encode events of linguistic nature through discrete categorical labels and parametric systems, such as TILT (147) and INTSINT (150) that aim only at providing a configurational description of the macroscopic pitch contour without any specific linguistic interpretation. While TILT and INTSINT are based on numerical and symbolic parameterizations of the pitch contour and hence are more or less language independent, ToBI requires expert human knowledge for the characterization of prosodic events in each language (e.g., Spanish ToBI (169), Japanese ToBI (170)). In contrast, the gross categorical descriptions within the ToBI framework offer a level of uncertainty in the human annotation to be incorporated into the labeling scheme and hence provide some generalization, considering that prosodic structure is highly speaker dependent. They also provide more general-purpose description of prosodic events encompassing acoustic correlates of pitch, duration and energy compared to TILT and INTSINT that exclusively model the pitch contour. Furthermore, the availability of large prosodically labeled corpora with manual ToBI annotations, such as the Boston University (BU) Radio News Corpus (167) and Boston Directions Corpus (BDC) (168), offer a convenient and standardized avenue to design and evaluate automatic ToBI-based prosody labeling algorithms.

Several linguistic theories have been proposed to represent the grouping of prosodic constituents (146; 171; 172). In the simplest representation, prosodic phrasing constituents can be grouped into *word*, *minor phrase*, *major phrase* and *utterance* (141). The ToBI break index representation (146) uses indices between 0 and 4 to denote the perceived disjuncture between each pair of words, while the perceptual labeling system described in (171) represents a superset of prosodic constituents by using labels between 0 and 6. In general, these representations are mediated by rhythmic and segmental analysis in the orthographic tier and associate each word with an appropriate index.

In this chapter, we evaluate our automatic prosody algorithm on the Boston University Radio News Corpus and Boston Directions Corpus, both of which are hand annotated with ToBI labels. We perform both prominence and phrase structure detection that are characterized within the ToBI framework through the following parallel tiers: (i) a tone tier, and (ii) a break-index tier. We provide a brief description of the ToBI annotation scheme and the associated characterization of prosodic prominence and phrasing by the parallel tiers in the following section.

8.1.1 ToBI annotation scheme

The Tones and Break Indices (ToBI) (146) framework consists of four parallel tiers that reflect the multiple components of prosody. Each tier consists of discrete categorical symbols that represent

Table 8.1: ToBI label mapping used in experiments. The decomposition of labels is illustrated for pitch accents, phrasal tones and break indices

ToBI Labels	**Intermediate Mapping**	**Coarse Mapping**
H* L+H*	High	**accent**
!H*, H+!H* L+!H*,L*+!H	Downstepped	
L* L*+H	Low	
,?,X*?	Unresolved	
L-L%,!H-L%,H-L% H-H% L-H% %?,X%?,%H	Final Boundary tone	**btone**
L-,H-,!H- -X?,-?	Intermediate Phrase (IP) boundary	
<,>,no label	**none**	**none**
0	0	**NB**
1,1-,1p	1	
2,2-,2p	2	
3,3-,3p	3	**B**
4,4-	4	

Table 8.2: Summary of previous work on pitch accent and boundary tone detection (coarse mapping). Level denotes the orthographic level (word or syllable) at which the experiments were performed. The results of Hasegawa-Johnson et. al and our work are directly comparable as the experiments are performed on identical dataset

Authors	Algorithm	Corpus	Level	Accuracy (%)	
				Pitch accent	**Boundary tone**
Wightman and Ostendorf (142)	HMM/CART	BU	syllable	83.0	77.0
Ross and Ostendorf (173)	HMM/CART	BU	syllable	87.7	66.9
Ananthakrishnan et al. (159)	Coupled HMM	BU	syllable	75.0	88.0
Gregory and Altun (161)	Conditional random fields	Switchboard	word	76.4	-
Nenkova et al. (174)	Decision Tree	Switchboard	word	76.6	-
Harper et al. (JHU Workshop) (175)	Decision Trees/ Random Forest	Switchboard	word	80.4	-
Hirschberg (155)	CART	BU	word	82.4	-
Wang and Hirschberg (176)	CART	ATIS	word	-	90.0
Ananthakrishnan et al. (159)	Coupled HMM	BU	word	79.5	82.1
Hasegawa Johnson et al. (160)	Neural networks/GMM	BU	word	84.2	93.0
Proposed work	Maximum entropy model	BU and BDC	word	86.0	93.1

prosodic events belonging to that particular tier[1]. A concise summary of the four parallel tiers is presented below. The reader is referred to (146) for a more comprehensive description of the annotation scheme.

- Orthographic tier: The orthographic tier contains the transcription of the orthographic words of the spoken utterance.
- Tone tier: Two types of tones are marked in the tonal tier: pitch events associated with intonational boundaries, *phrasal tones or boundary tones* and pitch events associated with accented syllables, *pitch accents*. The basic tone levels are high (H) and low (L), and are defined based on the relative value of the fundamental frequency in the local pitch range. There are a total of five pitch accents that lend prominence to the associated word: {H*, L*, L*+H, L+H*, H+!H*}. The phrasal tones are divided in two coarse categories, weak *intermediate phrase boundaries* {L-, H-} and *full intonational phrase boundaries* {L-L%, L-H%, H-H%, H-L%} that group together semantic units in the utterance.
- Break index tier: The break-index tier marks the perceived degree of separation between lexical items (words) in the utterance and is an indicator of prosodic phrase structure. Break indices range in value from 0 through 4, with 0 indicating no separation, or *cliticization*, and 4 indicating a full pause, such as at a sentence boundary. This tier is strongly correlated with phrase tone markings on the tone tier.
- Miscellaneous tier: This may include annotation of non-speech events such as disfluencies, laughter, etc.

The detailed representation of prosodic events in the ToBI framework however, suffers from the drawback that all the prosodic events are not equally likely and hence a prosodically labeled corpus

[1]On a variety of speaking styles, Pitrelli et al. (177) have reported inter-annotator agreements of 83-88%, 94-95% and 92.5%, respectively for pitch accent, boundary tone and break index detection within the ToBI annotation scheme

would consist of only a few instances of one event while comprising a majority of another. This in turn creates serious data sparsity problems for automatic prosody detection and identification algorithms. This problem has been circumvented to some extent by decomposing the ToBI labels into intermediate or coarse categories such as presence or absence of pitch accents, phrasal tones, etc., and performing automatic prosody detection on the decomposed inventory of labels. Such a grouping also reduces the effects of labeling inconsistency. A detailed illustration of the label decompositions is presented in Table 8.1. In this work, we use the coarse representation (presence versus absence) of pitch accents, boundary tones and break indices to alleviate the data sparsity and compare our results with previous work.

8.2 Related Work

In this section, we survey previous work in prominence and phrase break prediction with an emphasis on ToBI-based pitch accent, boundary tones and break index prediction. We present a brief overview of speech applications that have used such prosodic representations along with algorithms and their corresponding performance on the various prosody detection and identification tasks.

8.2.1 Pitch accent and boundary tone labeling

Automatic prominence labeling through pitch accents and boundary tones, has been an active research topic for over a decade. Wightman and Ostendorf (142) developed a decision-tree algorithm for labeling prosodic patterns. The algorithm detected phrasal prominence and boundary tones at the syllable level. Bulyko and Ostendorf (162) used a prosody prediction module to synthesize natural speech with appropriate pitch accents. Verbmobil (178) incorporated prosodic prominence into a translation framework for improved linguistic analysis and speech understanding.

Pitch accent and boundary tone labeling has been reported in many past studies (155; 159; 160). Hirschberg (155) used a decision-tree based system that achieved 82.4% speaker dependent accent labeling accuracy at the word level on the BU corpus using lexical features. Wang and Hirschberg (176) used a CART based labeling algorithm to achieve intonational phrase boundary classification accuracy of 90.0%. Ross and Ostendorf (173) also used an approach similar to (142) to predict prosody for a TTS system from lexical features. Pitch accent accuracy at the word-level was reported to be 82.5% and syllable-level accent accuracy was 87.7%. Hasegawa-Johnson et al. (160) proposed a neural network based syntactic-prosodic model and a gaussian mixture model based acoustic-prosodic model to predict accent and boundary tones on the BU corpus that achieved 84.2% accuracy in accent prediction and 93.0% accuracy in intonational boundary prediction. With syntactic information alone they achieved 82.7% and 90.1% for accent and boundary prediction, respectively. Ananthakrishnan and Narayanan (159) modeled the acoustic-prosodic information using a coupled hidden markov model that modeled the asynchrony between the acoustic streams. The pitch accent and boundary tone detection accuracy at the syllable level were 75% and 88% respectively. Yoon (179) has recently proposed memory-based learning approach and has reported accuracies of 87.78% and 92.23% for pitch accent and boundary tone labeling. The experiments were conducted on a subset of the BU corpus with 10,548 words and consisted of data from same speakers in the training and test set.

More recently, pitch accent labeling has been performed on spontaneous speech in the Switchboard corpus. Gregory and Atlun (161) modeled lexical, syntactic and phonological features using conditional random fields and achieved pitch accent detection accuracy of 76.4% on a subset of words in the Switchboard corpus. Ensemble machine learning techniques such as bagging and random forests on decision trees were used in the 2005 JHU Workshop (175) to achieve pitch accent

detection accuracy of 80.4%. The corpus used was a prosodic database consisting of spontaneous speech from the Switchboard corpus (180). Nenkova et al. (174) have reported a pitch accent detection accuracy of 76.6% on a subset of the Switchboard corpus using a decision tree classifier.

Our proposed maximum entropy discriminative model outperforms previous work on prosody labeling on the BU and BDC corpora. On the BU corpus, with syntactic information alone we achieve pitch accent and boundary tone accuracy of 85.2% and 91.5% on the same training and test sets used in (160; 181). These results are statistically significant by a difference of proportions test[2]. Further, the coupled model with both acoustic and syntactic information results in accuracies of 86.0% and 93.1% respectively. The pitch accent improvement is statistically significant compared to results reported in (181) by a difference of proportions test. On the BDC corpus, we achieve pitch accent and boundary tone accuracies of 79.8% and 90.3%. The proposed work uses speech and language information that can be reliably and easily extracted from the speech signal and orthographic transcription. It does not rely on any hand-coded features (174) or prosody labeled lexicons (160). The results of previous work on pitch accent and boundary tone detection on the BU corpus are summarized in Table 8.2.

8.2.2 Prosodic phrase break labeling

Automatic intonational phrase break prediction has been addressed mainly through rule-based systems developed by incorporation of rich linguistic rules, or, data-driven statistical methods that use labeled corpora to induce automatic labeling information (142; 166; 182; 183). Typically, syntactic information like POS tags, syntactic structure (parse features), as well as acoustic correlates like duration of preboundary syllables, boundary tones, pauses and f0 contour have been used as features in automatic detection and identification of intonational phrase breaks. Algorithms based on machine learning techniques such as decision trees (142; 166; 184), HMM (182) or combination of these (183) have been successfully used for predicting phrase breaks from text and speech.

Automatic detection of phrase breaks has been addressed mainly from the intent of incorporating the information in text-to-speech systems (166; 182), to generate appropriate pauses and lengthening at phrase boundaries. Phrase breaks have also been modeled from the interest of their utility in resolving syntactic ambiguity (153; 184; 185). Intonational phrase break prediction is also important in speech understanding (142) where the recognized utterance needs to be interpreted correctly.

One of the first efforts in automatic prosodic phrasing was presented by Ostendorf and Wightman (142). Using the seven level break index proposed in (171), they achieved an accuracy of 67% for exact identification and 89% correct identification within ± 1. They used a simple decision tree classifier for this task. Wang and Hirschberg (176) have reported an overall accuracy of 81.7% in detection of phrase breaks through a CART based scheme. Ostendorf and Veilleux (185) achieved 70% accuracy for break correct prediction, while, Taylor and Black (182), using their HMM based phrase break prediction based on POS tags have demonstrated 79.27% accuracy in correctly detecting break indices. Sun and Applebaum (183) have reported F-score of 77% and 93% on break and non-break prediction. Recently, ensemble machine learning techniques such as bagging and random forests that combined decision tree classifiers were used at the 2005 JHU workshop (175) to perform automatic break index labeling. The classifiers were trained on spontaneous speech (180) and resulted in break index detection accuracy of 83.2%. Kahn et al. (153) have also used prosodic break index labeling to improve parsing. Yoon (179) has reported break index accuracy of 88.06% in a three-way classification between break indices using only lexical and syntactic features.

[2]Results at a level ≤ 0.001 were considered significant

Table 8.3: Summary of previous work on break index detection (coarse mapping). Detection is performed at word-level for all experiments

Authors	Algorithm	Corpus	Accuracy (%) **Break index**
Wightman and Ostendorf (142)	HMM/CART	BU	84.0
Ostendorf and Veilleux (185)	HMM/CART	ATIS	70.0
Wang and Hirschberg (176)	CART	ATIS	81.7
Taylor and Black (182)	HMM	Spoken English corpus	79.2
Sun and Applebaum (183)	CART	BU	85.2
Harper et al. (JHU Workshop) (175)	Decision Trees/ Random Forest	Switchboard	83.2
Proposed work	Maximum entropy model	BU and BDC	84.0-87.5

We achieve a break index accuracy of 83.95% and 87.18% on the BU and BDC corpora using lexical and syntactic information alone. Our combined maximum entropy acoustic-prosodic model achieves a break index detection accuracy of 84.01% and 87.58%, respectively on the two corpora. The results from previous work are summarized in Table 8.3.

8.3 Maximum Entropy discriminative model for prosody labeling

Discriminatively trained classification techniques have emerged as one of the dominant approaches for resolving ambiguity in many speech and language processing tasks. Models trained using discriminative approaches have been demonstrated to out-perform generative models as they directly optimize the conditional distribution without modeling the distribution of all the underlying variables. The maximum entropy approach can model the uncertainty in labels in typical NLP tasks and hence is desirable for prosody detection due to the inherent ambiguity in the representation of prosodic events through categorical labels. A preliminary formulation of the work in this section was presented by the authors in (186; 187).

We model the prosody prediction problem as a classification task as follows: given a sequence of words w_i in an utterance $W = \{w_1, \cdots, w_n\}$, the corresponding syntactic information sequence $S = \{s_1, \cdots, s_n\}$ (for e.g., parts-of-speech, syntactic parse, etc.), a set of acoustic-prosodic features $A = \{\mathbf{a}_1, \cdots, \mathbf{a}_n\}$, where $\mathbf{a}_i = (a_i^1, \cdots, a_i^{t_{w_i}})$ is the acoustic-prosodic feature vector corresponding to word w_i with a frame length of t_{w_i} and a prosodic label vocabulary $\mathcal{L} = \{l_1, \cdots, l_V\}$, the best prosodic label sequence $L^* = \{l_1, l_2, \cdots, l_n\}$ is obtained as follows,

$$L^* = \arg\max_L P(L|W, S, A) \tag{8.1}$$

We approximate the string level global classification problem, using conditional independence assumptions, to a product of local classification problems as shown in Eq.(8.3). The classifier is then used to assign to each word a prosodic label conditioned on a vector of local contextual features

comprising the lexical, syntactic and acoustic information.

$$L^* = \arg\max_L P(L|W,S,A) \tag{8.2}$$

$$\approx \arg\max_L \prod_{i=1}^{n} p(l_i|w_{i-k}^{i+k}, s_{i-k}^{i+k}, \mathbf{a}_{i-k}^{i+k}) \tag{8.3}$$

$$= \arg\max_L \prod_{i=1}^{n} p(l_i|\mathbf{\Phi}(W,S,A,i)) \tag{8.4}$$

where $\mathbf{\Phi}(W,S,A,i) = (w_{i-k}^{i+k}, s_{i-k}^{i+k}, \mathbf{a}_{i-k}^{i+k})$ is a set of features extracted within a bounded local context k. $\mathbf{\Phi}(W,S,A,i)$ is shortened to $\mathbf{\Phi}$ in the rest of the section.

To estimate the conditional distribution $P(l_i|\mathbf{\Phi})$ we use the general technique of choosing the maximum entropy (maxent) distribution that estimates the average of each feature over the training data (188). This can be written in terms of the Gibbs distribution parameterized with weights λ_l, where l ranges over the label set and V is the size of the prosodic label set. Hence,

$$P(l_i|\mathbf{\Phi}) = \frac{e^{\lambda_{l_i}.\mathbf{\Phi}}}{\sum_{l=1}^{V} e^{\lambda_l.\mathbf{\Phi}}} \tag{8.5}$$

To find the global maximum of the concave function in Eq.(8.5), we use Sequential L1-Regularized Maxent algorithm (SL1-Max) (189). Compared to Iterative Scaling (IS) and gradient descent procedures, this algorithm results in faster convergence and provides L1-regularization as well as efficient heuristics to estimate the regularization meta-parameters. We use the machine learning toolkit LLAMA (190) to estimate the conditional distribution using maxent. LLAMA encodes multiclass maxent as binary maxent to increase the training speed and to scale the method to large data sets. We use here V one-versus-other binary classifiers. Each output label l is projected onto a bit string, with components $b_j(l)$. The probability of each component is estimated independently:

$$P(b_j(l)|\mathbf{\Phi}) = \frac{e^{\lambda_j.\mathbf{\Phi}}}{e^{\lambda_j.\mathbf{\Phi}} + e^{\lambda_{\bar{j}}.\mathbf{\Phi}}} = \frac{1}{1 + e^{-(\lambda_j - \lambda_{\bar{j}}).\mathbf{\Phi}}} \tag{8.6}$$

$$1 - P(\bar{b}_j(l)|\mathbf{\Phi}) =$$

where $\lambda_{\bar{j}}$ is the parameter vector for $\bar{b}_j(y)$. Assuming the bit vector components to be independent, we have,

$$P(l_i|\mathbf{\Phi}) = \prod_{j=1}^{V} P(b_j(l_i)|\mathbf{\Phi}) \tag{8.7}$$

Therefore, we can decouple the likelihoods and train the classifiers independently. In this work, we use the simplest and most commonly studied code, consisting of V one-versus-others binary components. The independence assumption states that the output labels or classes are independent.

8.4 Lexical, syntactic and acoustic features

In this section, we describe the lexical, syntactic and acoustic features that we use in our maximum entropy discriminative modeling framework. We use only features that are derived from the local

context of the text being tagged, referred to as static features here on. One would have to perform a Viterbi search if the preceding prediction context were to be added. Using static features is especially suitable for performing prosody labeling in lockstep with recognition or dialog act detection, as the prediction can be performed incrementally instead of waiting for the entire utterance or dialog to be decoded.

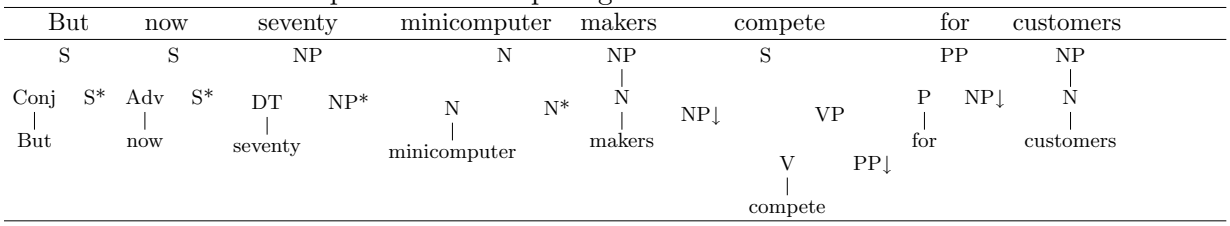
Table 8.4: Lexical, syntactic and acoustic features used in the experiments. The acoustic features were obtained over 10ms frame intervals

Category	Features used
Lexical features	Word identity (3 previous and next words)
Syntactic features	POS tags (3 previous and next words) Supertags (3 previous and next words) function/content word distinction (3 previous and next words)
Acoustic features	Speaker normalized f0 contour (+delta+acceleration) Speaker normalized energy contour (+delta+acceleration)

8.4.1 Lexical and syntactic features

The lexical features used in our modeling framework are simply the words in a given utterance. The BU and BDC corpora that we use in our experiments are automatically labeled (and hand-corrected) with part-of-speech (POS) tags. The POS inventory is the same as the Penn treebank which includes 47 POS tags: 22 open class categories, 14 closed class categories and 11 punctuation labels. We also automatically tagged the utterances using the AT&T POS tagger. The POS tags were mapped into function and content word categories[3] and were added as a discrete feature.

Table 8.5: Illustration of the supertags generated for a sample utterance in BU corpus. Each sub-tree in the table corresponds to one supertag.



In addition to the POS tags, we also annotate the utterance with Supertags (191). Supertags encapsulate predicate-argument information in a local structure. They are the elementary trees of Tree-Adjoining Grammars (TAGs) (192). Similar to part-of-speech tags, supertags are associated with each word of an utterance, but provide much richer information than part-of-speech tags, as illustrated in the example in Table V. Supertags can be composed with each other using substitution and adjunction operations (192) to derive the predicate-argument structure of an utterance.

[3]Function and content word features were obtained through a look-up table based on POS

There are two methods for creating a set of supertags. One approach is through the creation of a wide coverage English grammar in the lexicalized tree adjoining grammar formalism, called XTAG (193). An alternate method for creating supertags is to employ rules that decompose the annotated parse of a sentence in Penn Treebank into its elementary trees (194; 195). This second method for extracting supertags results in a larger set of supertags. For the experiments presented in this chapter, we employ a set of 4,726 supertags extracted from the Penn Treebank.

There are many more supertags per word than part-of-speech tags, since supertags encode richer syntactic information than part-of-speech tags. The task of identifying the correct supertag for each word of an utterance is termed as supertagging (191). Different models for supertagging that employ local lexical and syntactic information have been proposed (196). For the purpose of this chapter, we use a Maximum Entropy supertagging model that achieves a supertagging accuracy of 87% (197)[4].

While there have been previous attempts to employ syntactic information for prosody labeling (184; 198), which mainly exploited the local constituent information provided in a parse structure, supertags provide a different representation of syntactic information. First, supertags localize the predicate and its arguments within the same local representation (e.g. *give* is a di-transitive verb) and this localization extends across syntactic transformations (relativization, passivization, wh-extraction), i.e., there is a different supertag for each of these transformations for each of the argument positions. Second, supertags also factor out recursion from the predicate-argument domain. Thus modification relations are specified through separate supertags as shown in Table V. For this chapter we use the supertags as labels, even though there is a potential to exploit the internal representation of supertags as well as the dependency structure between supertags as demonstrated in (199). Table 8.5 shows the supertags generated for a sample utterance in the BU corpus.

8.4.2 Acoustic-prosodic features

The BU corpus contains the corresponding acoustic-prosodic feature file for each utterance. The f0 and RMS energy (e) of the utterance along with features for distinction between voiced/unvoiced segment, cross-correlation values at estimated f0 value and ratio of first two cross correlation values are computed over 10 msec frame intervals. The pitch values for unvoiced regions are smoothed using linear interpolation. In our experiments, we use these values rather than computing them explicitly which is straightforward with most audio processing toolkits. Both the energy and the f0 levels were range normalized (znorm) with speaker specific means and variances. Delta and acceleration coefficients were also computed for each frame. The final feature vector has 6 dimensions comprising f0, Δf0, Δ^2f0, e, Δe, Δ^2e per frame.

We model the frame level continuous acoustic-prosodic observation sequence as a discretized sequence through quantization (see Figure 8.1). We perform this on the normalized pitch and energy extracted from the time segment corresponding to each word. The quantized acoustic stream is then used as a feature vector. For this case, Eq.(8.3) becomes,

$$L^* \approx \arg\max_L \prod_i^n p(l_i|\mathbf{\Phi}) = \arg\max_L \prod_i^n p(l_i|\mathbf{a}_i) \tag{8.8}$$

where $\mathbf{a}_i = (a_i^1, \cdots, a_i^{t_{w_i}})$, the acoustic-prosodic feature vector corresponding to word w_i with a frame length of t_{w_i}.

[4]The model is trained to disambiguate among the supertags of a word by using the lexical and part-of-speech features of the word and of six words in the left and right context of that word. The model is trained on 1 million

Normalized pitch contour values:
-3.2595 0.2524 0.3634 0.2558 0.1960 0.1728 0.1845

Quantization (precision 2):
-3.25 0.25 0.36 0.25 0.19 0.17 0.18

Feature input to maxent classifier:
[(-3.25)], [(0.25),(0.25|-3.25)], ... , [(0.18),(0.18|0.17),(0.18|0.17,0.19)]

Figure 8.1: Illustration of the quantized feature input to the maxent classifier. "|" denotes feature input conditioned on preceding values in the acoustic-prosodic sequence

The quantization while being lossy, reduces the vocabulary of the acoustic-prosodic features, and hence offers better estimates of the conditional probabilities. The quantized acoustic-prosodic cues are then modeled using the maximum entropy model described in Section 8.3. The n-gram representation of quantized continuous features is similar to representing the acoustic-prosodic features with a piecewise linear fit as done in the tilt intonational model (147). Essentially, we leave the choice of appropriate representations of the pitch and energy features to the maximum entropy discriminative classifier, which integrates feature selection during classification.

Table 8.6: Statistics of Boston University Radio News and Boston Directions corpora used in experiments

Corpus statistics	BU				BDC			
	f2b	**f1a**	**m1b**	**m2b**	**h1**	**h2**	**h3**	**h4**
# Utterances	165	69	72	51	10	9	9	9
# words (w/o punc)	12608	3681	5058	3608	2234	4127	1456	3008
# pitch accents	6874	2099	2706	2016	1006	1573	678	1333
# boundary tones (w IP)	3916	1059	1282	1023	498	727	361	333
# boundary tones (w/o IP)	2793	684	771	652	308	428	245	216
# breaks (level 3 & above)	3710	1034	11721	1016	434	747	197	542

The proposed scheme of quantized n-gram prosodic features as input to the maxent classifier is different from previous work (200). Shriberg et al. (200) have proposed N-grams of Syllable-based Nonuniform Extraction Region Features (SNERF-grams) for speaker recognition. In their approach, they extract a large set of prosodic features such as maximum pitch, mean pitch, minimum pitch, durations of syllable onset, coda, nucleus, etc. and quantize these features by binning them. The resulting syllable-level features, for a particular bin resolution, are then modeled as either unigram (using current syllable only), bigram (current and previous syllable or pause) or trigram (current and previous two syllables or pauses). They use support vector machines (SVMs) for subsequent classification. Our framework, on the other hand, models the macroscopic prosodic contour in its entirety by using n-gram feature representation of the quantized prosodic feature sequence. This representation coupled with the strength of the maxent model to handle large feature sets and in avoiding overtraining through regularization makes our scheme attractive for capturing characteristic pitch movements associated with prosodic events.

words of supertag annotated text.

Table 8.7: Baseline classification results of pitch accents and boundary tones (in %) using Festival and AT&T Natural Voices speech synthesizer

Corpus	Speaker Set	Prediction Module	Accuracy	
			Pitch accent	**Boundary tone**
BU	Entire Set	**Chance**	54.33	81.14
		Lexical stress	72.64	-
		AT&T Natural Voices	81.51	89.10
		Festival	69.55	89.54
	Hasegawa-Johnson et al. set	**Chance**	56.53	82.88
		Lexical stress	74.10	-
		AT&T Natural Voices	81.73	89.67
		Festival	68.65	90.21
BDC	Entire Set	**Chance**	57.60	88.90
		Lexical stress	67.42	-
		AT&T Natural Voices	68.49	84.90
		Festival	64.94	85.17

8.5 Experimental Evaluation

8.5.1 Data

All the experiments reported in this chapter are performed on the Boston University (BU) Radio News Corpus (167) and the Boston Directions Corpus (BDC) (168), two publicly available speech corpora with manual ToBI annotations intended for experiments in automatic prosody labeling. The BU corpus consists of broadcast news stories including original radio broadcasts and laboratory simulations recorded from seven FM radio announcers. The corpus is annotated with orthographic transcription, automatically generated and hand-corrected part-of-speech tags and automatic phone alignments. A subset of the corpus is also hand annotated with ToBI labels. In particular, the experiments in this chapter are carried out on 4 speakers similar to (181), 2 males and 2 females referred to hereafter as **m1b**, **m2b**, **f1a** and **f2b**. The BDC corpus is made of elicited monologues produced by subjects who were instructed to perform a series of direction-giving tasks. Both spontaneous and read versions of the speech are available for four speakers **h1**, **h2**, **h3** and **h4** with hand-annotated ToBI labels and automatic phone alignments, similar to the BU corpus. Table 8.6 shows some of the statistics of the speakers in the BU and BDC corpora.

In all our prosody labeling experiments we adopt a leave-one-out speaker validation similar to the method in (160) for the four speakers with data from one speaker for testing and those from the other three for training. For the BU corpus, speaker **f2b** was always used in the training set since it contains the most data. In addition to performing experiments on all the utterances in BU corpus, we also perform identical experiments on the train and test sets reported in (181) which is referred to as Hasegawa-Johnson et al. set.

8.6 Pitch accent and boundary tone labeling

In this section, we present pitch accent and boundary tone labeling results obtained through the proposed maximum entropy prosody labeling scheme. We first present some baseline results, followed by the description of results obtained from our classification framework.

8.6.1 Baseline Experiments

We present three baseline experiments. One is simply based on chance where the majority class label is predicted. The second is a baseline only for pitch accents derived from the lexical stress obtained through look-up from a pronunciation lexicon labeled with stress. Finally, the third baseline is obtained through prosody detection in current off-the-shelf speech synthesis systems. The baseline using speech synthesis systems is comparable to our proposed model that uses lexical and syntactic information alone. For experiments using acoustics, our baseline is simply chance.

Table 8.8: Classification results (%) of pitch accents and boundary tones for different syntactic representations. Classifiers with cardinality V=2 learned either accent or btone classification, classifiers with cardinality V=4 classified accent and btone simultaneously. The variable (k) controlling the length of the local context was set to $k = 3$

Corpus	Speaker Set	Syntactic features	V=2		V=4	
			Pitch accent	**Boundary tone**	**Pitch accent**	**Boundary tone**
BU	Entire Set	correct POS tags	84.75	91.39	84.60	91.34
		POS tags	83.71	90.52	83.50	90.36
		POS + supertags	84.59	91.34	84.48	91.22
	Hasegawa-Johnson et al. set	correct POS tags	85.22	91.33	85.03	91.29
		POS tags	83.91	90.14	83.72	90.04
		POS + supertags	84.95	91.21	84.85	91.24
BDC	Entire Set	POS + supertags	79.81	90.28	79.57	89.76

8.6.1.1 Acoustic baseline (chance)

The simplest baseline we use is chance, which refers to the majority class label assignment for all tokens. The majority class label for pitch accents is presence of a pitch accent (**accent**) and that for boundary tone is absence (**none**).

8.6.1.2 Prosody labels derived from lexical stress

Pitch accents are usually carried by the stressed syllable in a particular word. Lexicons with phonetic transcription and lexical stress are available in many languages. Hence, one can use these lexical stress markers within the syllables and evaluate the correlation with pitch accents. Even when the lexicon has a closed vocabulary, letter-to-sound rules can be derived from it for unseen words. For each word carrying a pitch accent, we find the particular syllable where the pitch accent occurs from the manual annotation. For the same syllable, we assign a pitch accent based on the presence or absence of a lexical stress marker in the phonetic transcription. The CMU pronunciation lexicon was used for predicting lexical stress through simple lookup. Lexical stress for out-of-vocabulary words was predicted through a CART based letter-to-sound rule derived from the pronunciation lexicon. The results are presented in Table 8.7.

8.6.1.3 Prosody labels predicted using TTS systems

We perform prosody prediction using two off-the-shelf speech synthesis systems, namely, AT&T NV speech synthesizer and Festival. The AT&T NV speech synthesizer (201) is a half phone speech synthesizer. The toolkit accepts an input text utterance and predicts appropriate ToBI pitch accent and boundary tones for each of the selected units (in this case, a pair of phones) from the database. The toolkit uses a rule-based procedure to predict the ToBI labels from lexical information (155). We reverse mapped the selected half phone units to words, thus obtaining the ToBI labels for each word in the input utterance. The pitch accent labels predicted by the toolkit are $L_{\mathbf{accent}} \epsilon \{\mathbf{H}*, \mathbf{L}*, \mathbf{none}\}$ and the boundary tones are $L_{\mathbf{btone}} \epsilon \{\mathbf{L\text{-}L}\%, \mathbf{H\text{-}H}\%, \mathbf{L\text{-}H}\%, \mathbf{none}\}$.

Festival (202) is an open-source unit selection speech synthesizer. The toolkit includes a CART-based prediction system that can predict ToBI pitch accents and boundary tones for the input text utterance. The pitch accent labels predicted by the toolkit are $L_{\mathbf{accent}} \epsilon \{\mathbf{H}*, \mathbf{L}+\mathbf{H}*, !\mathbf{H}*, \mathbf{none}\}$ and the boundary tones are $L_{\mathbf{btone}} \epsilon \{\mathbf{L\text{-}L}\%, \mathbf{H\text{-}H}\%, \mathbf{L\text{-}H}\%, \mathbf{none}\}$. The prosody labeling results obtained through both the speech synthesis engines are presented in Table 8.7.

8.6.2 Maximum entropy pitch accent and boundary tone classifier

In this section, we present results of our maximum entropy pitch accent and boundary tone classification. We first present a maximum entropy syntactic-prosodic model that uses only lexical and syntactic information for prosody detection, followed by a maximum entropy acoustic-prosodic model that uses an n-gram feature representation of the quantized acoustic-prosodic observation sequence.

8.6.2.1 Maximum entropy syntactic-prosodic model

The maximum entropy syntactic-prosodic model uses only lexical and syntactic information for prosody labeling. Our prosodic label inventory consists of $L_{\mathbf{accent}} \epsilon \{\mathbf{accent}, \mathbf{none}\}$ for pitch accents, and $L_{\mathbf{btone}} \epsilon \{\mathbf{btone}, \mathbf{none}\}$ for boundary tones. Such a framework is beneficial for text-to-speech synthesis that relies on lexical and syntactic features derived predominantly from the input text to synthesize natural sounding speech with appropriate prosody. The results are presented in Table 8.8. In Table 8.8, correct POS tags refer to hand-corrected POS tags present in the BU corpus release and POS tags refers to parts-of-speech tags predicted automatically.

Prosodic prominence and phrasing can also be viewed as joint events occurring simultaneously. Previous work by (142) suggests that a joint labeling approach may be more beneficial in prosody labeling. In this scenario, we treat each word to have one of the four labels $l_i \epsilon \mathcal{L} = \{$**accent-btone, accent-none, none-btone, none-none**$\}$. We trained the classifier on the joint labels and then computed the error rates for individual classes. The joint modeling approach provides a marginal improvement in the boundary tone prediction but is slightly worse for pitch accent prediction.

8.6.2.2 Maximum entropy acoustic-prosodic model

We quantize the continuous acoustic-prosodic values by binning, and extract n-gram features from the resulting sequence. The quantized acoustic-prosodic n-gram features are then modeled with a maxent acoustic-prosodic model similar to the one described in section 5. Finally, we append the syntactic and acoustic features to model the combined stream with the maxent acoustic-syntactic model, where the objective criterion for maximization is Eq.(8.1). The two streams of information were weighted in the combined maximum entropy model by performing optimization on the training

Table 8.9: Classification results of pitch accents and boundary tones (in %) with acoustics only, syntax only and acoustics+syntax using both our models. The syntax based results from our maximum entropy syntactic-prosodic classifier are presented again to view the results cohesively. In the table A = Acoustics, S = Syntax

Corpus	Speaker Set	Model	**Pitch accent**			**Boundary tone**		
			A	S	A+S	A	S	A+S
BU	Entire Set	Maxent acoustic model	80.09	84.60	84.63	84.10	91.36	91.76
		HMM acoustic model	70.58	84.60	85.13	71.28	91.36	92.91
	Hasegawa-Johnson et al. set	Maxent acoustic model	80.12	84.95	85.16	82.70	91.54	91.94
		HMM acoustic model	71.42	84.95	86.01	73.43	91.54	93.09
BDC	Entire Set	Maxent acoustic model	74.51	79.81	79.97	83.53	90.28	90.49
		HMM acoustic model	68.57	79.81	80.01	74.28	90.28	90.58

set (weights of 0.8 and 0.2 were used on the syntactic and acoustic vectors respectively). The pitch accent and boundary tone prediction accuracies for quantization performed by considering only the first decimal place is reported in Table 8.9. As expected, we found the classification accuracy to drop with increasing number of bins used in the quantization due to the small amount of training data. In order to compare the proposed maxent acoustic-prosodic model with conventional approaches such as HMMs, we also trained continuous observation density HMMs to represent pitch accents and boundary tones. This is presented in detail in the following section.

8.6.3 HMM acoustic-prosodic model

In this section, we compare the proposed maxent acoustic-prosodic model with a traditional HMM approach. HMMs have been demonstrated to capture the time-varying pitch patterns associated with pitch accents and boundary tones effectively (158; 159). We trained separate context independent HMMs with 3 state left-to-right topology with uniform segmentation. The segmentations need to be uniform due to lack of an acoustic-prosodic model trained on the features pertinent to our task to obtain forced segmentation. The acoustic observations of the HMM were unquantized acoustic-prosodic features described in Section 8.4.2. The label sequence was decoded using the Viterbi algorithm.

The final label sequence using the maximum entropy syntactic-prosodic model and the HMM based acoustic-prosodic model was obtained by combining the syntactic and acoustic probabilities. Essentially, the prosody labeling task reduces to the following:

$$\begin{aligned} L^* &= \arg\max_L P(L|A,W) \\ &= \arg\max_L P(L|W).P(A|L,W) \\ &\approx \arg\max_L P(L|\Phi(W)).P(A|L)^\gamma \end{aligned} \tag{8.9}$$

where $\Phi(W)$ is the syntactic feature encoding of the word sequence W. The first term in Eq.(8.9) corresponds to the probability obtained through our maximum entropy syntactic model. The second term in Eq.(8.9), computed by an HMM corresponds to the probability of the acoustic data stream which is assumed to be dependent only on the prosodic label sequence. γ is a weighting factor to adjust the weight of the two models.

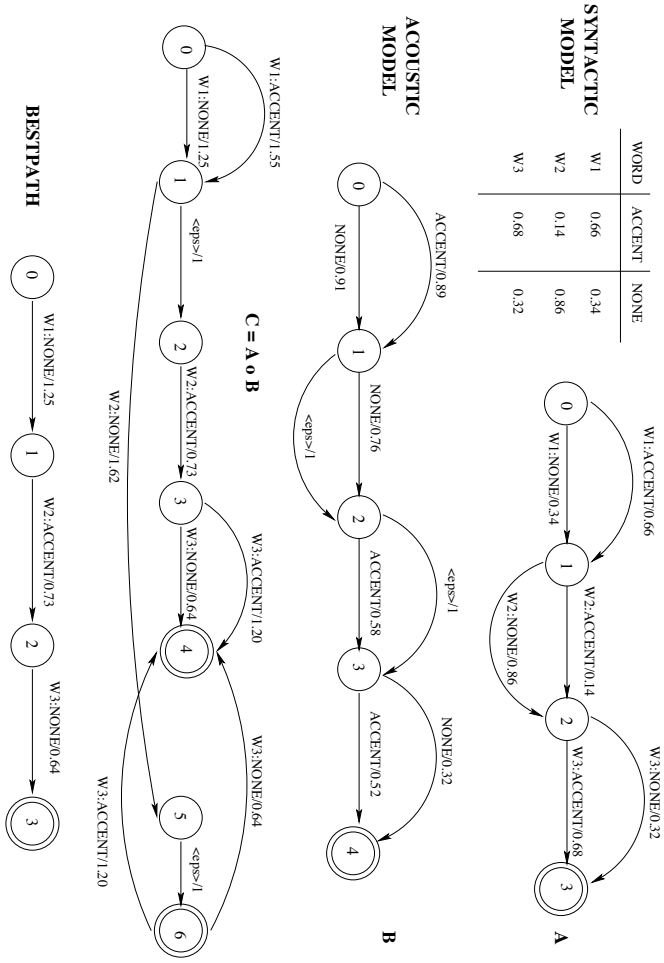


Figure 8.2: Illustration of the FST composition of the syntactic and acoustic lattices and resulting best path selection. The syntactic-prosodic maxent model produces the syntactic lattice and the HMM acoustic-prosodic model produces the acoustic lattice.

The syntactic-prosodic maxent model outputs a posterior probability for each class per word. We formed a lattice out of this structure and composed it with the lattice generated by the HMM acoustic-prosodic model. The best path was chosen from the composed lattice through a Viterbi search. The procedure is illustrated in Figure 8.2. The acoustic-prosodic probability $P(A|L, W)$ was raised by a power of γ to adjust the weighting between the acoustic and syntactic model. The value of γ was chosen as 0.008 and 0.015 for pitch accent and boundary tone respectively, by tuning on the training set. The results of the HMM acoustic-prosodic model and the coupled model are

shown in Table 8.9. The weighted maximum entropy syntactic-prosodic model and HMM acoustic-prosodic model performs the best in pitch accent and boundary tone classification. We conjecture that the generalization provided by the acoustic HMM model is complementary to that provided by the maximum entropy model, resulting in slightly better accuracy when combined together as compared to that of a combined maxent-based acoustic and syntactic model.

8.7 Prosodic break index labeling

We presented pitch accent and boundary tone labeling results using our proposed maximum entropy classifier in the previous section. In the following section, we address phrase structure detection by performing automatic break index labeling within the ToBI framework. Prosodic phrase break prediction has been especially useful in text-to-speech (182) and sentence disambiguation (184; 185) applications, both of which rely on prediction based on lexical and syntactic features. We follow the same format as the prominence labeling experiments, presenting baseline experiments followed by our maximum entropy syntactic and acoustic classification schemes. All the experiments are performed on the entire BU and BDC corpora.

8.7.1 Baseline Experiments

We present baseline experiments, both chance and break index labeling results using an off-the-shelf speech synthesizer. The AT&T Natural Voices speech synthesizer does not have a prediction module for prosodic break prediction and hence we present results from using the Festival (202) speech synthesizer alone. Festival speech synthesizer produces simple binary break presence or absence distinction, as well as more detailed ToBI-like break index prediction.

8.7.1.1 Break index prediction in Festival

Festival can predict break index at the word level based on the algorithm presented in (182). The toolkit can predict both, ToBI like break values ($L_{\mathbf{tobi\_break}} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$) and simple presence versus absence ($L_{\mathbf{binary\_break}} \in \{\mathbf{B}, \mathbf{NB}\}$). Only lexical and syntactic information is used in this prediction without any acoustics. Baseline classification results are presented in Table 8.10.

8.7.2 Maximum Entropy model for break index prediction

8.7.2.1 Syntactic-prosodic model

The maximum entropy syntactic-prosodic model uses only lexical and syntactic information for prosodic break index labeling. Our prosodic label inventory consists of $L_{\mathbf{tobi\_break}} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$ for ToBI based break indices, and $L_{\mathbf{binary\_break}} \in \{\mathbf{B}, \mathbf{NB}\}$ for binary break versus no-break distinction. The $\{\mathbf{B}, \mathbf{NB}\}$ categorization was obtained by grouping break indices $\mathbf{0}, \mathbf{1}, \mathbf{2}$ into **NB** and $\mathbf{3}, \mathbf{4}$ into **B** (146). The classifier is then applied for break index labeling as described in Section 8.6.2.1 for the pitch accent prediction. We assume knowledge of sentence boundary through the means of punctuation in all the reported experiments.

8.7.2.2 Acoustic-prosodic model

Prosodic break index prediction is typically used in text-to-speech systems and syntactic parse disambiguation. Hence, the lexical and syntactic features are crucial in the automatic modeling of

Table 8.10: Classification results of break indices (in %) with syntax only, acoustics only and acoustics+syntax using the maximum entropy classifier. In the table A = Acoustics, S = Syntax

	ToBI break indices					**B/NB**				
Corpus	Chance	Festival	A	S	A+S	Chance	Festival	A	S	A+S
BU	61.25	64.22	64.73	72.32	72.90	71.91	77.58	73.98	83.95	84.01
BDC	60.01	66.56	58.95	69.25	69.81	82.26	82.31	75.94	87.18	87.58

these prosodic events. Further, they are defined at the word level and do not demonstrate a high degree of correlation with specific pitch patterns. We thus use only the maximum entropy acoustic-prosodic model described in Section 8.6.2.2. The combined maximum entropy acoustic-syntactic model is then similar to Eq.(8.2), where the prosodic label sequence is conditioned on the words, POS tags, supertags and quantized acoustic-prosodic features. A binary flag indicating the presence or absence of a pause before and after the current word was also included as a feature. The results of the maximum entropy syntactic, acoustic and acoustic-syntactic model for break index prediction are presented in Table 8.10. The maxent syntactic-prosodic model achieves break index detection accuracies of 83.95% and 87.18% on the BU and BDC corpora. The addition of acoustics to the lexical and syntactic features does not result in a significant improvement in detection accuracy. In these experiments, we used only pitch and energy features and did not use duration features such as rhyme duration, duration of final syllable, etc., used in (142). Such features require both phonetic alignment and syllabification and therefore are difficult to obtain in speech applications that require automatic prosody detection to be performed in lockstep. Additionally, in the context of TTS systems and parsers, the proposed maximum entropy syntactic-prosodic model for break index prediction performs with high accuracy compared to previous work.

8.8 Discussion

The automatic prosody labeling presented in this work is based on ToBI-based categorical prosody labels but is extendable to other prosodic representation schemes such as IViE (149) or INTSINT (150). The experiments are performed on decompositions of the original ToBI labels into binary classes. However, with the availability of sufficient training data, we can overcome data sparsity and provide more detailed prosodic event detection (refer to Table 8.1). We use acoustic features only in the form of pitch and energy contour for pitch accent and boundary tone detection. Durational features, which are typically obtained through forced alignment of the speech signal at the phone level in typical prosody detection tasks have not been considered in this work. We concentrate only on the energy and pitch contour that can be robustly obtained from the speech signal. However, our framework is readily amenable to the addition of new features. We provide discussions on the prominence and phrase structure detection presented in sections 8.6 and 8.7 below.

8.8.1 Prominence prediction

The baseline experiment with lexical stress obtained from a pronunciation lexicon for prediction of pitch accent yields substantially higher accuracy than chance. This could be particularly useful in resource-limited languages where prosody labels are usually not available but one has access to a reasonable lexicon with lexical stress markers. Off-the-shelf speech synthesizers like Festival and

AT&T speech synthesizer have utilities that perform reasonably well in pitch accent and boundary tone prediction. AT&T speech synthesizer performs better than Festival in pitch accent prediction while the latter performs better in boundary tone prediction. This can be attributed to better rules in the AT&T synthesizer for pitch accent prediction. Boundary tones are usually highly correlated with punctuation and Festival seems to capture this well. However, both these synthesizers generate a high degree of false alarms.

The maximum entropy model syntactic-prosodic proposed in section 8.6.2.1 outperforms previously reported results on pitch accent and boundary tone classification. Much of the gain comes from the strength of the maximum entropy modeling in capturing the uncertainty in the classification task. Considering the inter-annotator agreement for ToBI labels is only about 81% for pitch accents and 93% for boundary tones, the maximum entropy framework is able to capture the uncertainty present in manual annotation. The supertag feature offers additional discriminative information over the part-of-speech tags (also demonstrated by Rambow and Hirschberg (199)).

The maximum entropy acoustic-prosodic model discussed in section 8.6.2.2 performs well in isolation compared to the traditional HMM acoustic-prosodic model. This is a simple method and the quantization resolution can be adjusted based on the amount of data available for training. However, the model performs with slightly lower accuracy when combined with the syntactic features compared to the combined maxent syntactic-prosodic and HMM acoustic-prosodic model. We conjecture that the generalization provided by the acoustic HMM model is complementary to that provided by the maximum entropy acoustic model, resulting in slightly better accuracy when combined with the maxent syntactic model compared the maxent acoustic-syntactic model. We attribute this behavior to better smoothing offered by the HMM compared to the maxent acoustic model. We also expect this slight difference would not be noticeable with a larger data set.

The weighted maximum entropy syntactic-prosodic model and HMM acoustic-prosodic model performs the best in pitch accent and boundary tone classification. The classification accuracies are comparable to the inter-annotator agreement for the ToBI labels. Our HMM acoustic-prosodic model is a generative model and does not assume the knowledge of word boundaries in predicting the prosodic labels as in previous approaches (142; 155; 160). This makes it possible to have true parallel prosody prediction during speech recognition. However, the incorporation of word boundary knowledge, when available, can aid in improved detection accuracies (203). This is also true in the case of our maxent acoustic-prosodic model that assumes word segmentation information. The weighted approach also offers flexibility in prosody labeling for either speech synthesis or speech recognition. While the syntactic-prosodic model would be more discriminative for speech synthesis, the acoustic-prosodic model is more appropriate for speech recognition.

8.8.2 Phrase structure prediction

The baseline results from Festival speech synthesizer are relatively modest for the break index prediction and only slightly better than chance. The break index prediction module in the synthesizer is mainly based on punctuation and parts-of-speech tag information and hence does not provide a rich set of discriminative features. The accuracies reported on the BU corpus are substantially higher compared to chance than those reported on the BDC corpus. We found that the distribution of break indices was highly skewed in the BDC corpus and the corpus also does not contain any punctuation markers. Our proposed maximum entropy break index labeling with lexical and syntactic information alone achieves 83.95% and 87.18% accuracy on the BU and BDC corpora. The syntactic model can be used in text-to-speech synthesis and sentence disambiguation (for parsing) applications. We also envision the use of prosodic breaks in speech translation by aiding in the construction of improved phrase translation tables.

8.9 Summary, conclusions, and future work

In this chapter, we described a maximum entropy discriminative modeling framework for automatic prosody labeling. We applied the proposed scheme to both prominence and phrase structure detection within the ToBI annotation scheme. The proposed maximum entropy syntactic-prosodic model alone resulted in pitch accent and boundary tone accuracies of 85.2% and 91.5% on training and test sets identical to (181). As far as we know, these are the best results on the BU and BDC corpus using syntactic information alone and a train-test split that does not contain the same speakers. We have also demonstrated the significance of our approach by setting reasonable baseline from out-of-the-box speech synthesizers and by comparing our results with prior work. Our combined maximum entropy syntactic-prosodic model and HMM acoustic-prosodic model performs the best with pitch accent and boundary tone labeling accuracies of 86.0% and 93.1% respectively. The results of collectively using both syntax and acoustic within the maximum entropy framework are not far behind at 85.2% and 92% respectively. The break index detection with the proposed scheme is also promising with detection accuracies ranging from 84-87%. The inter-annotator agreement for pitch accent, boundary tone and break index labeling on the BU corpus (167) are 81-84%, 93% and 95%, respectively. The accuracies of 80-86%, 90-93.1% and 84-87% achieved with the proposed framework for the three prosody detection tasks are comparable to the inter-labeler agreements. In summary, the experiments of this chapter demonstrate the strength of using a maximum entropy discriminative model for prosody prediction. Our framework is also suitable for integration into state-of-the-art speech applications.

The supertag features in this work were used as categorical labels. The tags can be unfolded and the syntactic dependencies and structural relationship between the nodes of the supertags can be exploited further as demonstrated in (199). We plan to use these more refined features in future work. As a continuation of our work, we have integrated our prosody labeler in a dialog act tagging scenario and we have been able to achieve modest improvements (204). We are also working on incorporating our automatic prosody labeler in a speech-to-speech translation framework. Typically, state-of-the-art speech translation systems have a source language recognizer followed by a machine translation system. The translated text is then synthesized in the target language with prosody predicted from text. In this process, some of the critical prosodic information present in the source data is lost during translation. With reliable prosody labeling in the source language, one can transfer the prosody to the target language (this is feasible for languages with phrase level correspondence). The prosody labels by themselves may or may not improve the translation accuracy but they provide a framework where one can obtain prosody labels in the target language from the speech signal rather than depending only on a lexical prosody prediction module in the target language.

Bibliography

[1] "Fisher-related conversational style web collection," http://ssli.ee.washington.edu/projects/ears/WebData/w

[2] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 1596–1608, 2006.

[3] E. Rich, "Users are individuals: individualizing user models," *International Journal of Human-Computer Studies*, vol. 51, no. 2, pp. 323–338, 1999.

[4] X. Yang, K. Wang, and S. Shamma, "Auditory representations of acoustic signals," vol. 38, no. 2, pp. 824–839, 1992.

[5] M. Lapata and F. Keller, "Web-based models for natural language processing," in *ACM Transactions on Speech and Language Processing*, 2005.

[6] F. Keller, M. Lapata, and O. Ourioupina, "Using the web to overcome data sparseness," in *Proceedings of EMNLP*, 2002.

[7] P. Resnik and N. A. Smith, "The web as a parallel corpus," *Computational Linguistics*, vol. 29, pp. 349–380, 2003.

[8] D. Graff, "English Gigaword," 2003, LDC Catalog ID: LDC2003T05.

[9] T. Ng, M. Ostendorf, M.-Y. Hwang, M. Siu, I. Bulyko, and X. Lei, "Web-data augmented language model for mandarin speech recognition," in *Proceedings of ICASSP*, 2005.

[10] R. Sarikaya, A. Gravano, and Y. Gao, "Rapid language model development using external resources for new spoken dialog domains," in *Proceedings of ICASSP*, 2005.

[11] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," in *Journal of Machine Learning*, 2000.

[12] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005, http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

[13] A. Sethy, P. G. Georgiou, and S. Narayanan, "Text data acquisition for domain-specific language models," in *Proceedings of EMNLP*, 2006.

[14] S. Narayanan and P. G. et al., "Transonics: A speech to speech system for english-persian interactions," in *Proceedings of IEEE ASRU*, 2003.

[15] B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig, M. Westphal, H. Schulz, and A. Soneiro, "The IBM 2006 speech transcription system for European parliamentary speeches," in *Proceedings of ICSLP*, 2006.

[16] T. Misu and T. Kawahara, "A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts," in *Proceedings of ICSLP*, 2006.

[17] A. Sethy, P. Georgiou, and S. Narayanan, "Building topic specific language models from web-data using competitive models," in *Proceedings of Eurospeech*, 2005.

[18] K. Weilhammer, M. N. Stuttlem, and S. Young, "Bootstrapping language models for dialogue systems," in *Proceedings of ICSLP*, 2006.

[19] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. Yu., "Building text classifiers using positive and unlabeled examples," in *Proceedings of ICDM*, 2003.

[20] A. Stolcke, "Entropy-based pruning of backoff language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[21] L. Lee, "Measures of distributional similarity," in *37th Annual Meeting of the Association for Computational Linguistics*, 1999, pp. 25–32.

[22] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, 1996.

[23] A. Ratnaparkhi, "A maximum entropy part-of-speech tagger," in *Proceedings of EMNLP*, 1996.

[24] "TC-STAR: Technology and corpora for speech to speech translation," http://www.tc-star.org.

[25] B. Pellom, "SONIC: the university of colorado continuous speech recognizer," 2001.

[26] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based csr corpus," in *Workshop on Speech and Natural Language*, 1992.

[27] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text," in *Proceedings of LREC*, 2004.

[28] D. Graff, K. Walker, and A. Canavan, "Switchboard-2 Phase II," 1999, LDC Catalog ID: LDC99S79.

[29] K. Seymore, S. Chen, S.-J. Doh, E. Gouvea, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern, and E. Thayer, "The 1997 CMU Sphinx-3 English Broadcast News transcription system," in *Proceedings of the 1998 DARPA Speech Recognition Workshop*, 1998.

[30] E. Ettelaie, S. Gandhe, P. Georgiou, K. Knight, D. Marcu, S. Narayanan, D. Traum, and R. Belvin, "Transonics: a practical speech-to-speech translator for English-Farsi medical dialogues," in *Proceedings of the ACL*, 2005.

[31] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of ICSLP*, 2002.

[32] A. Sethy, B. Ramabhadran, and S. Narayanan, "Measuring convergence in language model estimation using relative entropy," in *Proceedings of ICSLP*, 2004.

[33] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proceedings of HLT*, 2003.

[34] O.Cetin and A. Stolcke, "Language modeling in the ICSI-SRI spring 2005 meeting speech recognition evaluation," in *ICSI Technical Report TR-05-006*, 2005.

[35] R. Iyer, M. Ostendorf, and M. Meteer, "Analyzing and predicting language model improvements," in *Proceedings of ASRU*, 1997.

[36] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Proceedings of Eurospeech*, 1999.

[37] Y. Z. Fei Huang and S. Vogel, "Mining key phrase translations from web corpora," in *Proceedings of EMNLP*, 2005.

[38] J. Bellegarda, "Large vocabulary speech recognition with multispan statistical language models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, 2000.

[39] B.J.Hsu and J.Glass, "Style and topic language model adaptation using HMM-LDA," in *Proceedings of EMNLP*, 2006.

[40] T. Brants and A. Franz, "Web 1T 5-gram Version 1," 2006, LDC Catalog ID : LDC2006T13.

[41] H. Hermansky and N. Morgan, "Rasta processing of speech," vol. 2, no. 4, pp. 578–589, October 1994.

[42] K. Wang and S. Shamma, "Self-normalization and noise robustness in early auditory representations," vol. 2, no. 3, pp. 421–435, 1994.

[43] W. Jeon and B.-H. Juang, "A study of auditory modeling and processing for speech signals," in *IEEE Int. Conf. Acoust., Speech Signal Proc.*, vol. 1, Pensylvania, USA, 2005, pp. 929–932.

[44] S. Ravindran, D. V. Anderson, and M. Slaney, "Improving the noise-robustness of mel-frequency cepstral coefficients for speech processing," in *SAPA*, Pittsburgh, PA, September 2006.

[45] N. Mesgarani, S. Shamma, and M. Slaney, "Speech discrimination based on multiscale spectro-temporal modulations," in *IEEE Int. Conf. Acoust., Speech Signal Proc.*, Montreal, Canada, May 2004.

[46] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, Paris, France, September 2000.

[47] S. Young. (1989) Hidden markov model toolkit (HTK). [Online]. Available: http://htk.eng.cam.ac.uk/

[48] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*. New York, NY: Oxford University Press, 1999.

[49] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, NY: Wiley-Interscience, 2001.

[50] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. New York, NY: John Wiley and Sons Inc., 2000.

[51] S. Narayanan, S. Ananthakrishnan, R. Belvin, E. Ettaile, S. Ganjavi, P. G. Georgiou, C. M. Hein, S. Kadambe, K. Knight, D. Marcu, H. E. Neely, N. Srinivasamurthy, D. Traum, and D. Wang, "Transonics: A speech to speech system for English-Persian interactions," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2003.

[52] B. Zhou, Y. Gao, J. Sorensen, D. Dechelotte, and M. Picheny, "A hand-held speech-to-speech translation system," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2003.

[53] K. Precoda and R. J. Podesva, "What will people say? speech system design and language/cultural differences," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2003.

[54] A. W. Black, R. D. Brown, R. Frederking, R. Singh, J. Moody, and E. Steinbrecher, "TONGUES: rapid development of a speech-to-speech translation system," in *Proceedings of the 2nd International Conference on Human Language Technology Research (HLT)*, March 2002. [Online]. Available: citeseer.ist.psu.edu/black02tongues.html

[55] W. Eckert, E. Levin, and R. Pieraccini, "User modeling for spoken dialogue system evaluation," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)*, 1997.

[56] K. Georgila, J. Henderson, and O. Lemon, "Learning user simulations for information state update dialogue systems," in *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech/Interspeech)*, 2005.

[57] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

[58] K. Komatani, S. Ueno, T. Kawahara, and H. G. Okuno, "Flexible guidance generation using user model in spoken dialogue systems," *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 256–263, 2003.

[59] D. Litman and S. Pan, "Empirically evaluating an adaptable spoken dialog system," in *Proceedings of the 7th International Conference on User Modeling (UM)*, 1999.

[60] J. E. Pitkow, H. Schuetze, T. A. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. M. Breuel, "Personalized search," vol. 45, no. 9, 2002, pp. 50–55.

[61] G. W. Bauer, "Interface for user/agent interaction," in *U.S. Patent 5877759, Mar 2*, 1999.

[62] H. Yan and T. Selker, "Context-aware office assistant," in *Proceedings of International Conference on Intelligent User Interfaces*, 2000.

[63] I. Zukerman and D. W. Albrech, "Predictive statistical models for user modeling," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 5–18, 2001.

[64] A. Kuenzer, C. Schlick, F. Ohmann, L. Schmidt, and H. Luczak, "An empirical study of dynamic bayesian networks for user modeling," in *Proceedings of the UM 2001 Workshop on Machine Learning for User Modeling*, 2001.

[65] J. A. Hall, D. L. Roter, and N. R. Katz, "Meta-analysis of correlates of provider behavior in medical encounters," *Medical Care*, vol. 26, no. 7, pp. 657–675, 1988.

[66] D. L. Roter and J. A. Hall, "Studies of doctor-patient interaction," *Annual Review of Public Health*, vol. 10, pp. 163–180, 1989.

[67] K. Pitschke, "User modeling for domains without explicit design theories," in *Proceedings of the 4th International Conference on User Modeling (UM)*, Hyannis, MA, 1994.

[68] E. Manavoglu, D. Pavlov, and C. L. Giles, "Probabilistic user behavior models," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003.

[69] E. Rich, "User modeling via stereotypes," *International Journal of Cognitive Science*, vol. 3, pp. 329–354, 1979.

[70] S. Carberry, J. Chu-Carroll, and S. Elzer, "Constructing and utilizing a model of user preferences in collaborative consultation dialogues," *Computational Intelligence*, vol. 15, no. 3, pp. 185–217, 1999.

[71] C. Conati, A. Gertner, and K. Vanlehn, "Using bayesian networks to manage uncertainty in student modeling," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 371–417, 2002.

[72] H. Prendinger, J. Mori, and M. Ishizuka, "Recognizing, modeling, and responding to users' affective states," in *Proceedings of the 10th International Conference of User Modeling (UM)*, 2005.

[73] A. Kobsa and W. Pohl, "The user modeling shell system bgp-ms," *User Modeling and User-Adapted Interaction*, vol. 4, no. 2, pp. 59–106, 1995.

[74] B. Pakucs, "Sesame: A framework for personalized and adaptive speech interfaces," in *Proceedings of the EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management*, 2003.

[75] V. Tsiriga and M. Virvou, "A framework for the initialization of student models in web-based intelligent tutoring systems," *User Modeling and User-Adapted Interaction*, vol. 14, no. 4, pp. 289–316, 2004.

[76] J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd, "Analysis of user behavior under error conditions in spoken dialogs," in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 2002.

[77] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Noth, "How to find trouble in communication," *Speech Communication*, vol. 40, no. 1-2, pp. 117–143, 2003.

[78] K. Jokinen and K. Kanto, "User expertise modelling and adaptativity in a speech-based e-mail system," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[79] C. A. Kamm, D. J. Litman, and M. A. Walker, "From novice to expert: The effect of tutorials on user expertise with spoken dialog systems," in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998.

[80] A. Kobsa, "Modeling the user's conceptual knowledge in bgp-ms, a user modeling shell system," *Computational Intelligence*, vol. 6, pp. 193–208, 1990.

[81] B. Grawemeyer and R. Cox, "A bayesian approach to modelling users' information display preferences," in *Proceedings of the 10th International Conference of User Modeling (UM)*, 2005.

[82] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning - Special issue on learning with probabilistic representations*, vol. 29, no. 2-3, pp. 131–163, 1997.

[83] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 3, 1997.

[84] J. Sheinvald, B. Dom, and W. Niblack, "A modeling approach to feature selection," in *Proceedings of the 10th International Conference on Pattern Recognition*, 1990.

[85] K. Zechner and A. Waibel, "Minimizing word error rate in textual summaries of spoken language," in *Proceedings of NAACL-ANLP-2000*, 2000. [Online]. Available: citeseer.ist.psu.edu/zechner00minimizing.html

[86] P. Prodanov and A. Drygajlo, "Bayesian networks based multi-modality fusion for error handling in human robot dialogues under noisy conditions," *Speech Communication*, vol. 45, no. 3, pp. 231–248, 2005.

[87] M. Walker, D. Litman, C. Kamm, and A. Abella, "PARADISE: A general framework for evaluating spoken dialogue agents," in *Proceedings of The Association for Computational Linguistics (ACL/EACL)*, 1997.

[88] J. Tian, J. Nurminen, and I. Kiss, "Optimal subset selection from text databases," in *Proceedings of 30th Anniversary IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[89] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[90] B. Merialdo, "Tagging English text with a probabilistic model," *Computational Linguistics*, vol. 20, pp. 155—171, 1994.

[91] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[92] J. Bilmes, "On soft evidence in Bayesian networks," University of Washington Dept. of Electrical Engineering, Tech. Rep. UWEETR-2004-00016, 2004.

[93] M. Johnson, "Why doesn't EM find good HMM POS-taggers?" in *Proc. EMNLP*, 2007.

[94] A. Clark, "Unsupervised language acquisition: Theory and practice," Ph.D. dissertation, COGS, University of Sussex, 2001.

[95] M. Banko and R. C. Moore, "Part of speech tagging in context," in *Proc. COLING*, 2004.

[96] S. M. Reynolds and J. A. Bilmes, "Part-of-speech tagging using virtual evidence and negative training," in *Proc. HLT-EMNLP*, 2005.

[97] A. Ratnaparkhi, "A maximum entropy part-of-speech tagger," in *Proc. EMNLP*, 1996.

[98] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. HLT-NAACL*, 2003.

[99] A. Haghighi and D. Klein, "Protoype-driven learning for sequence models," in *Proc. HLT-NAACL*, 2006.

[100] N. A. Smith and J. Eisner, "Contrastive estimation: Training log-linear models on unlabeled data," in *Proc. ACL*, 2005.

[101] S. Goldwater and T. L. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," in *Proc. ACL*, 2007.

[102] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, 2007.

[103] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley-Interscience, 1991.

[104] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001.

[105] J. Bilmes and G. Zweig, "The Graphical Models Toolkit: An open source software system for speech and time-series processing," in *Proc. ICASSP*, 2002.

[106] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, 1993.

[107] N. A. Smith, "Novel estimation methods for unsupervised discovery of latent structure in natural language text," Ph.D. dissertation, Johns Hopkins University, 2006.

[108] A. Clark, "Combining distributional and morphological information for part of speech induction," in *Proc. EACL*, 2003.

[109] J. Terken, T. Hermes, M. Ostendorf, and N. Campbell, *Prosody: Theory and Experiment.* Kluwer Academic Press, 2000, ch. 4,9,10.

[110] R. Kompe, *Prosody in Speech Understanding Systems.* Springer-Verlag, 1997.

[111] E. Shriberg and A. Stolcke, "Prosody modeling for automatic speech recognition and understanding," in *Proceedings of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, 2001, pp. 13–16.

[112] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard scheme for labeling prosody," in *Proceedings of the International Conference on Spoken Language Processing*, 1992, pp. 867–869.

[113] M. Beckman and G. Elam, *Guidelines for ToBI Labeling*, http://www.ling.ohio-state.edu/research/phonetics/E_ToBI.

[114] C. Wightman, "ToBI or not ToBI?" in *Proceedings of the Speech Prosody Conference*, 2002, pp. 25–30.

[115] D. Hirst and A. D. Cristo, *Intonation Systems: A Survey of Twenty Languages*, D. Hirst and A. D. Cristo, Eds. Cambridge University Press, 1998.

[116] *ToBI: The Ohio State University Department of Linguistics*, The Ohio State University, http://www.ling.ohio-state.edu/∼tobi, 1999.

[117] E. Grabe, F. Nolan, and K. Farrar, "IViE - A comparative transcription system for intonational variation in English," in *Proceedings of the International Conference on Spoken Language Processing*, 1998.

[118] P. Taylor, "The TILT intonation model," in *Proceedings of the International Conference on Spoken Language Processing*, vol. 4, 1998, pp. 1383–1386.

[119] C. Wightman and M. Ostendorf, "Automatic labeling of prosodic patterns," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 469–481, 1994.

[120] K. Ross and M. Ostendorf, "Prediction of abstract prosodic labels for speech synthesis," *Computer Speech and Language*, vol. 10, pp. 155–185, 1996.

[121] A. Syrdal, J. Hirschberg, J. McGory, and M. Beckman, "Automatic ToBI prediction and alignment to speed manual labeling of prosody," *Speech Communication*, vol. 33, no. 135-151, 2001.

[122] K. Chen, M. Hasegawa-Johnson, and A. Cohen, "An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2004, pp. 509–512.

[123] S. Ananthakrishnan and S. Narayanan, "An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2005, pp. 269–272.

[124] M. Ostendorf, P. Price, and S. Shattuck-Hufnagel, "The Boston University Radio News Corpus," 1995.

[125] D. Kahn, "Syllable-based generalizations in English phonology," Ph.D. dissertation, University of Massachusetts, 1976.

[126] C. Wightman, "Segmental durations in the vicinity of prosodic phrase boundaries," *Journal of the Acoustical Society of America*, 1992.

[127] S. Arnfield, "Prosody and syntax in corpus-based analysis of spoken english," Ph.D. dissertation, The University of Leeds School of Computer Studies, 1994.

[128] J. R. Quinlan, "Induction of decision trees," in *Machine Learning*. Kluwer Academic Press, 1986, vol. 1, pp. 81–106.

[129] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.

[130] W. Mendenhall and T. Sincich, *Statistics for Engineering and the Sciences*, 4th ed. Prentice-Hall, 1995.

[131] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[132] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[133] J. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proceedings of HLT/NAACL*, 2003.

[134] *The CMU Pronunciation Dictionary*, http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[135] A. Stolcke, "SRILM - An extensible language modeling toolkit," in *Proceedings of the International Conference of Spoken Language Processing*, 2002.

[136] M. Mohri and M. Riley, "Weighted finite-state transducers in speech recognition (tutorial)," in *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[137] G.-A. Levow, "Unsupervised and semi-supervised learning of tone and pitch accent," in *Proceedings of HLT/NAACL*, June 2006.

[138] S. Ananthakrishnan and S. Narayanan, "Combining acoustic, lexical, and syntactic evidence for automatic unsupervised prosody labeling," in *Proceedings of the International Conference on Spoken Language Processing*, 2006, pp. 297–300.

[139] M. Hasegawa-Johnson, J. Cole, C. Shih, K. Chen, A. Cohen, S. Chavarria, H. Kim, T. Yoon, S. Borys, and J.-Y. Choi, "Speech recognition models of the interdependence among syntax, prosody and segmental acoustics," in *Proceedings of HLT/NAACL*, 2004.

[140] S. Ananthakrishnan and S. Narayanan, "Improved speech recognition using acoustic and lexical correlates of pitch accent in a N-Best rescoring framework," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Honolulu, Hawaii, April 2007.

[141] I. Lehiste, *Suprasegmentals*. Cambridge, MA: MIT Press, 1970.

[142] C. W. Wightman and M. Ostendorf, "Automatic labeling of prosodic patterns," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 469–481, 1994.

[143] P. Koehn, S. Abney, J. Hirschberg, and M. Collins, "Improving intonational phrasing with syntactic information," in *Proceedings of ICASSP*, 2000.

[144] M. Steedman, "Information structure and the syntax-phonology interface," *Linguistic inquiry*, vol. 31, no. 4, pp. 649–689, 2000.

[145] M. Liberman and A. Prince, "On stress and linguistic rhythm," *Linguistic Inquiry*, vol. 8, no. 2, pp. 249–336, 1977.

[146] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *Proceedings of ICSLP*, 1992, pp. 867–870.

[147] P. Taylor, "The tilt intonation model," in *Proc. ICSLP*, vol. 4, 1998, pp. 1383–1386.

[148] H. Fujisaki and K. Hirose, "Modelling the dynamic characteristics of voice fundamental frequency with application to analysis and synthesis of intonation," in *Proceedings of 13th International Congress of Linguists*, 1982, pp. 57–70.

[149] E. Grabe, F. Nolan, and K. Farrar, "IViE - a comparative transcription system for intonational variation in English," in *Proceedings of ICSLP*, Sydney, Australia, 1998.

[150] D. J. Hirst, N. Ide, and J. Vronis, "Coding fundamental frequency patterns for multilingual synthesis with INTSINT in the MULTEXT project," in *Proceedings of the 2nd ESCA/IEEE Workshop on Speech Synthesis*, September 1994, pp. 77–81.

[151] E. E. Shriberg, R. A. Bates, and A. Stolcke, "A prosody-only decision-tree model for disfluency detection," in *Proc. Eurospeech 97*, Rhodes, Greece, 1997.

[152] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1526–1540, September 2006.

[153] J. G. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf, "Effective use of prosody in parsing conversational speech," in *Proceedings of HLT/EMNLP*, 2005.

[154] E. Shriberg, R. Bates, A. Stolcke, P. Taylor, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Van Ess-Dykema, "Can prosody aid the automatic classification of dialog acts in conversational speech?" *Language and Speech*, vol. 41, no. 3-4, pp. 439–487, 1998.

[155] J. Hirschberg, "Pitch accent in context: Predicting intonational prominence from text," *Artificial Intelligence*, vol. 63, no. 1-2, 1993.

[156] P. Shimei and K. McKeown, "Word informativeness and automatic pitch accent modeling," in *In Proceedings of EMNLP/VLC*, College Park, Maryland, 1999.

[157] X. Sun, "Pitch accent prediction using ensemble machine learning," in *Proc. of ICSLP*, 2002.

[158] A. Conkie, G. Riccardi, and R. C. Rose, "Prosody recognition from speech utterances using acoustic and linguistic based models of prosodic events," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 523–526.

[159] S. Ananthakrishnan and S. Narayanan, "An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model," in *In Proceedings of ICASSP*, Philadelphia, PA, March 2005.

[160] M. Hasegawa-Johnson, K. Chen, J. Cole, S. Borys, S.-S. Kim, A. Cohen, T. Zhang, J.-Y. Choi, H. Kim, T.-J. Yoon, and S. Chavara, "Simultaneous recognition of words and prosody in the boston university radio speech corpus," *Speech Communication*, vol. 46, pp. 418–439, 2005.

[161] M. Gregory and Y. Altun, "Using conditional random fields to predict pitch accent in conversational speech," in *42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[162] I. Bulyko and M. Ostendorf, "Joint prosody prediction and unit selection for concatenative speech synthesis," in *Proc. of ICASSP*, 2001.

[163] X. Ma, W. Zhang, Q. Shi, W. Zhu, and L. Shen, "Automatic prosody labeling using both text and acoustic information," in *Proceedings of ICASSP*, vol. 1, April 2003, pp. 516–519.

[164] P. Taylor, S. King, S. Isard, and H. Wright, "Intonation and dialogue context as constraints for speech recognition," *Language and Speech*, vol. 41, no. 34, pp. 493–512, 2000.

[165] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, pp. 339–373, Sep. 2000.

[166] J. Hirschberg and P. Prieto, "Training intonational phrasing rules automatically for English and Spanish text-to-speech," *Speech Commun.*, vol. 18, no. 3, pp. 281–290, 1996.

[167] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University Radio News Corpus," Boston University, Technical Report ECS-95-001, March 1995.

[168] J. Hirschberg and C. Nakatani, "A prosodic analysis of discourse segments in direction-giving monologues," in *Proceedings of the 34th conference on Association for Computational Linguistics*, 1996, pp. 286–293.

[169] M. E. Beckman, M. Diaz-Campos, J. T. McGory, and T. A. Morgan, "Intonation across spanish, in the tones and break indices framework," *Probus*, vol. 14, pp. 9–36.

[170] M. E. Beckman and J. B. Pierrehumbert, "Intonational structure in Japanese and English," *Phonology Yearbook*, vol. 3, pp. 255–309.

[171] P. J. Price, M. Ostendorf, S. Shattuck-Hufnagel, and C. Fong, "The use of prosody in syntactic disambiguation," *The Journal of the Acoustical Society of America*, vol. 90, no. 6, pp. 2956–2970, 1991. [Online]. Available: http://link.aip.org/link/?JAS/90/2956/1

[172] C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. J. Price, "Segmental durations in the vicinity of prosodic phrase boundaries," *J. of the Acoustical Society of America*, vol. 91, no. 3, pp. 1707–1717, 1992.

[173] K. Ross and M. Ostendorf, "Prediction of abstract prosodic labels for speech synthesis," *Computer Speech and Language*, vol. 10, pp. 155–185, Oct. 1996.

[174] A. Nenkova, J. Brenier, A. Kothari, S. Calhoun, L. Whitton, D. Beaver, and D. Jurafsky, "To memorize or to predict: Prominence labeling in conversational speech," in *Proceedings of NAACL-HLT 2007*, 2007.

[175] M. Harper, B. Dorr, B. Roark, J. Hale, Z. Shafran, Y. Liu, M. Lease, M. Snover, L. Young, R. Stewart, and A. Krasnyanskaya, "Parsing speech and structural event detection," JHU Summer Workshop, Tech. Rep., 2005.

[176] M. Q. Wang and J. Hirschberg, "Automatic classification of intonational phrase boundaries," *Computer Speech and Language*, vol. 6, pp. 175–196, 1992.

[177] J. F. Pitrelli, M. E. Beckman, and J. Hirschberg, "Evaluation of prosodic transcription labeling reliability in the tobi framework," in *Proceedings of ICSLP*, 1994, pp. 123–126.

[178] E. Nöth, A. Batliner, A. Kießling, R. Kompe, and H. Niemann, "VERBMOBIL: The use of prosody in the linguistic components of a speech understanding system," *IEEE Transactions on Speech and Audio processing*, vol. 8, no. 5, pp. 519–532, 2000.

[179] T.-J. Yoon, "Predicting prosodic boundaries using linguistic features," in *ICSA International Conference on Speech Prosody*, Dresden, Germany, 2006.

[180] M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, L. Carmichael, and W. Byrne, "A prosodically labeled database of spontaneous speech," in *ISCA Workshop on Prosody in Speech Recognition and Understanding*, 2001, pp. 119–121.

[181] K. Chen, M. Hasegawa-Johnson, and A. Cohen, "An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model," in *Proceedings of ICASSP*, 2004.

[182] A. W. Black and P. Taylor, "Assigning phrase breaks from part-of-speech sequences," in *Proc. of EUROSPEECH*, vol. 2, Rhodes, Greece, 1997, pp. 995–998.

[183] X. Sun and T. H. Applebaum, "Intonational phrase break prediction using decision tree and n-gram model," vol. 1, Aalborg, Denmark, 2001, pp. 537–540.

[184] N. M. Veilleux, M. Ostendorf, and C. W. Wightman, "Parse scoring with prosodic information," in *Proc. 1992 Intl. Conf. on Spoken Language Processing*, 1992, pp. 1605–1608.

[185] N. M. Veilleux and M. Ostendorf, "Prosody/parse scoring and its application in atis," in *HLT '93: Proceedings of the workshop on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 1993, pp. 335–340.

[186] V. K. Rangarajan Sridhar, S. Bangalore, and S. Narayanan, "Acoustic-syntactic maximum entropy model for automatic prosody labeling," in *Proceedings of IEEE/ACL Spoken Language Technology*, Aruba, Dec. 2006.

[187] ——, "Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework," in *Proceedings of NAACL-HLT*, 2007.

[188] A. Berger, S. D. Pietra, and V. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[189] M. Dudik, S. Phillips, and R. E. Schapire, "Performance guarantees for regularized maximum entropy density estimation," in *Proceedings of COLT*. Banff, Canada: Springer Verlag, 2004.

[190] P. Haffner, "Scaling large margin classifiers for spoken language understanding," *Speech Communication*, vol. 48, no. iv, pp. 239–261, 2006.

[191] S. Bangalore and A. K. Joshi, "Supertagging: An approach to almost parsing," *Computational Linguistics*, vol. 25, no. 2, 1999.

[192] A. Joshi and Y. Schabes, "Tree-adjoining grammars," in *Handbook of Formal Lanaguages and Automata*, A. Salomaa and G. Rozenberg, Ed. Springer-Verlag, Berlin, 1996.

[193] XTAG, "A lexicalized tree-adjoining grammar for English," University of Pennsylvania, http://www.cis.upenn.edu/ xtag/gramrelease.html, Tech. Rep., 2001.

[194] J. Chen and K. Vijay-Shanker, "Automated extraction of tags from the penn treebank," in *Proceedings of the 6th International Workshop on Parsing Technologies*, Trento, Italy, 2000.

[195] F. Xia, M. Palmer, and A. Joshi, "A uniform method of grammar extraction and its applications," in *Proceedings of Empirical Methods in Natural Language Processing*, 2000.

[196] S. Bangalore, A. Emami, and P. Haffner, "Factoring global inference by enriching local representations," AT&T Labs-Research, Tech. Rep., 2005.

[197] S. Bangalore and P. Haffner, "Classification of large label sets," in *Proceedings of the Snowbird Learning Workshop*, 2005.

[198] M. Hasegawa-Johnson, J. Cole, C. Shih, K. Chen, A. Cohen, S. Chavarria, H. Kim, T. Yoon, S. Borys, and J.-Y. Choi, "Speech recognition models of the interdependence among syntax, prosody, and segmental acoustics," in *Proceedings of HLT/NAACL, Workshop on Higher-Level Knowledge in Automatic Speech Recognition and Understanding*, May 2004.

[199] J. Hirschberg and O. Rambow, "Learning prosodic features using a tree representation," in *Proceedings of Eurospeech*, Aalborg, 2001, pp. 1175–1180.

[200] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, pp. 455–472, 2005.

[201] "AT&T Natural Voices speech synthesizer." http://www.naturalvoices.att.com.

[202] A. W. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system." http://festvox.org/festival, 1998.

[203] K. Chen, M. Hasegawa-Johnson, A. Cohen, S. Borys, S.-S. Kim, J. Cole, and J.-Y. Choi, "Prosody dependent speech recognition on radio news corpus of American English," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 232–245, 2006.

[204] V. K. Rangarajan Sridhar, S. Bangalore, and S. Narayanan, "Exploiting prosodic features for dialog act tagging in a discriminative modeling framework," in *Proceedings of ICSLP*, 2007.

[205] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, August 1991.

[206] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, pp. 269–311, 1997.

[207] R. C. Carrasco, "Accurate computation of the relative entropy between stochastic regular grammars," *RAIRO (Theoretical Informatics and Applications)*, vol. 31, no. 5, pp. 437–444, 1997.

[208] Takeo Kanade, "Immersion into visual media: New applications of image understanding," *IEEE Expert*, vol. 11(1), pp. 73-80, Feb. 1996.

[209] Yao Wang, Zhu Liu, and Jin-Cheng Huang, "Multimedia content analysis using both audio and visual cues," *IEEE Signal Process. Mag.* pp. 12-36, Nov. 2000.

[210] Lie Lu, Hong-Jiang Zhang, and Hao Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Speech Audio Process.*, vol. 10(7), pp. 504-516, Oct. 2002.

[211] Ying Li, Shrikanth S. Narayanan, and C.-C. J. Kuo, "Content-based movie analysis and indexing based on audiovisual cues," *IEEE Trans. Circuits Systems Video Tech.*, vol. 14(8), pp. 1073-1085, Aug. 2004.

[212] Klaus Krippendorffs, *Content analysis: An introduction to its methodology.* 2nd ed., Sage, 2004.

[213] Sue E. Tranter and Douglas A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14(5), pp. 1557-1565, Sept. 2006.

[214] Benchmark Tests: Rich Transcription. National Institute of Standards and Technology (NIST). [Online]. Available: http://www.nist.gov/speech/tests/rt/

[215] Douglas A. Reynolds and Pedro A. Torres-Carrasquillo, "Approaches and applications of audio diarization," *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 953-956, March 2005.

[216] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern classification.* 2nd edition, John Wiley & Sons, 2001.

[217] Douglas A. Reynolds and Pedro A. Torres-Carrasquillo, "The MIT Lincoln laboratory RT-04F diarization systems: Applications to broadcast news and telephone conversations," *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*, Nov. 2004.

[218] Rohit Sinha, Sue E. Tranter, Mark J. F. Gales, and Phil C. Woodland, "The Cambridge university March 2005 speaker diarisation system," *Proc. 9th European Conference on Speech Communication and Technology*, pp. 2437-2440, Mar. 2005.

[219] Xavier Anguera, Chuck Wooters, Barbara Peskin, and Mateu Aguilo, "Robust speaker diarization for meetings: ICSI RT06S meetings evaluation system," *Proc. 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, pp. 346-358, May 2006.

[220] David A. van Leeuwen and Marijn Huijbregts, "The AMI speaker diarization system for NIST RT06S meeting data," *Proc. 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, pp. 371-384, May 2006.

[221] Sylvain Meignier, Daniel Moraru, Corinne Fredouille, Jean-Francois Bonastre, and Laurent Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech Lang.*, vol. 20(2-3), pp. 303-330, July 2006.

[222] Claude Barras, Xuan Zhu, Sylvain Meignier, and Jean-Luc Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14(5), pp. 1505-1512, Sept. 2006.

[223] Gideon Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6(2), pp. 461-464, March 1978.

[224] Scott S. Chen and Panani S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp. 127-132, Feb. 1998.

[225] Herbert Gish, Man-Hung Siu, and Robin Rohlicek, "Segregation of speakers for speech recognition and speaker identification," *Proc. 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 873-876, May 1991.

[226] Kyu J. Han, Samuel Kim, and Shrikanth S. Narayanan, "Robust speaker clustering strategies to data source variation for improved speaker diarization," *2007 IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 262-267, Dec. 2007.

[227] Kyu J. Han and Shrikanth S. Narayanan. "A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system," *Proc. Interspeech 2007 - Eurospeech*, pp. 1853-1856, Aug. 2007.

[228] Daben Liu and Francis Kubala, "Fast speaker change detection for broadcast news transcription and indexing," *Proc. 6th European Conference on Speech Communication and Technology*, pp. 1031-1034, Sept. 1999.

[229] An Vandecatseye and Jean-Pierre Martens, "A fast, accurate and stream-based speaker segmentation and clustering algorithm," *Proc. Interspeech 2003 - Eurospeech*, pp. 941-944, Sept. 2003.

[230] Thomas. M. Cover and Joy. A. Thomas, *Elements of information theory.* John Wiley & Sons, 1991.

[231] Jitendra Ajmera and Chuck Wooters, "A robust speaker clustering algorithm," *2003 IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 411-416, Nov. 2003.

[232] Jitendra Ajmera, Iain McCowan, and Herve Bourlard, "Robust speaker change detection," *IEEE Signal Process. Lett.*, vol. 11(8), pp. 649-651, Aug. 2004.

[233] Douglas A. Reynolds and Richard C. Rose, "Robust text-independent speaker identification using Gaussian mixture models," *IEEE Trans. Speech Audio Process.*, vol. 3(1), pp. 72-83, Jan. 1995.

[234] Douglas A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Comm.*, vol. 17(1-2), pp. 91-108, Aug. 1995.

[235] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10(1-3), pp. 19-41, July 2000.

[236] Carlos Busso, Panayiotis G. Georgiou, and Shrikanth S. Narayanan. "Real-time monitoring of participants interaction in a meeting using audio-visual sensors," *Proc. 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 685-688, April 2007.

Appendix A

Generalization to back-off n-grams

For a given context $h = w_1..w_{n-1}$, let $S(h)$ be the set of words for which the probability estimate $p(w|h)$ is explicitly defined in the back-off n-gram model. Consider the n-gram probabilities $p(w|h)$ which lie in the complement set $S^c(h)$. The probability of these n-grams can be computed in terms of the probability of the back-off n-gram with history $h' = w_2..w_{n-1}$ as,

$$p(w|h) = b(h) * p(w|h')$$

where $b(h)$ is called the back-off weight. Given that $\sum_{w \in V} p(w|h) = 1$ it is easy to see that

$$b(h) = \frac{1 - \sum_{w \in S} p(w|h)}{1 - \sum_{w \in S} p(w|h')}$$

The primary problem in extending the data selection algorithm presented in Section 2.2 to back-off n-grams is the increase in computational complexity of calculating the relative entropy change resulting from changes in the back-off parameters. To keep the complexity tractable we developed a data selection scheme which enforces the back-off structure of the in-domain model on the n-gram model built from the selected data. Note that the assumption that the two models share the same back-off structure, does not limit the selection of data to n-gram histories seen in the in-domain data. By restricting the back-off structure for the model built from selected data, we fix whether we will update an n-gram estimate or modify the corresponding back-off weights. Other methods which can reduce the complexity include treating the model built from selected data as a non back-off ML model. We have not experimented with these alternate strategies.

We first describe a scheme for the fast computation of R.E. between two back-off n-gram language models which share the same back-off structure. We use the generalized derivation from (32) adapted to the case where the two language models have the same back-off structure. To keep the presentation of the algorithm simple, we will use the entropy model described in (13). This can be changed to the skew divergence model described for the unigram case described in Section 2.2 by adjusting the counts to include the in-domain model probability.

A.1 Fast Computation of Relative Entropy

We define the following symbols for the purpose of describing the R.E. computation:

w: The current word
h: The history $w_1..w_{n-1}$
h': The back-off history $w_2..w_{n-1}$

$b^p(h)$: The back-off weight for the p distribution for history h
$b^q(h)$: The back-off weight for the q distribution for history h
V: The vocabulary of the language model

The information theoretic measure of relative entropy rate (205) can be used to compare discrete Markovian distributions such as n-gram language models. Given two n-gram language models $p(w|h)$ and $q(w|h)$, the relative entropy rate at level n is defined as

$$R(n) = \sum_{h \in H} p(h) \sum_{w \in V} p(w|h) \ln \frac{p(w|h)}{q(w|h)} \tag{A.1}$$

where H is the set of all possible histories at level n.

In the rest of this discussion, we will refer to relative entropy rate as just relative entropy. Let us denote the conditional relative entropy (205) between the two n-gram distributions p and q for the history h with D_h. We have

$$D(h) = \sum_{w \in V} p(w|h) \ln \frac{p(w|h)}{q(w|h)} \tag{A.2}$$

We now divide the set of all possible histories (H) at level n into H_s for all h which exist as $n-1$ gram and have a back-off weight $\neq 1$ in the p or the q distribution. The complement set (H_s^c) will contain histories with a back-off weight of 1. H_s^c corresponds to histories not seen in either language model. Then R.E. at level n, $R(n)$ can be expressed as

$$\begin{aligned} R(n) &= \sum_{h \in H} p(h) D(h) \\ &= \sum_{h \in H_s} p(h) D(h) + \sum_{h \in H_s^c} p(h) D(h) \\ &= \sum_{h \in H} p(h) D(h') + \sum_{h \in H_s} p(h) D(h) - \sum_{h \in H_s} p(h) D(h') \end{aligned}$$

Since $h{=}w{\cdot}h'$, we can marginalize with respect to w

$$\begin{aligned} R(n) &= \sum_{w} \sum_{h'} p(h) D(h') + \sum_{h \in H_s} p(h) D(h) - \sum_{h \in H_s} p(h) D(h') \\ &= \sum_{h'} D(h') \sum_{w} p(h) + \sum_{h \in H_s} p(h) \left(D(h) - D(h') \right) \\ &= \sum_{h'} D(h') p(h') + \sum_{h \in H_s} p(h) \left(D(h) - D(h') \right) \\ &= R(n-1) + \sum_{h \in H_s} p(h) \left(D(h) - D(h') \right) \end{aligned}$$

Let us denote by $S(h)$ the set of words w, for history h, for which the n-gram $w|h$ is explicitly defined in the two LMs, p and q which share the same back-off structure. We use $S^c(h)$ to denote the complement of set $S(h)$. In (32), $D(h)$ is split into four terms depending on whether $w|h$ is explicitly defined in the p or the q distribution. When the two LMs have the same back-off structure we need to consider only two terms in the expansion of $D(h)$. We call these terms $T_1(h)$

and $T_4(h)$, to use the same notation as the derivation in (32). $T_1(h)$ corresponds to terms $p(w|h)$ and $q(w|h)$ which exist as explicit n-grams ($w \in S(h)$) and $T_4(h)$ corresponds to $p(w|h)$ and $q(w|h)$ which back-off ($w \in S^c(h)$). We have,

$$D(h) = T_1(h) + T_4(h) \tag{A.3}$$

$$\begin{aligned} T_1(h) &= \sum_{w \in S(h)} p(w|h) \ln \frac{p(w|h)}{q(w|h)} \\ T_4(h) &= \sum_{w \in S^c(h)} b^p(h) p(w|h') \ln \frac{b^p(h) p(w|h')}{b^q(h) q(w|h')} \\ &= b^p(h) \ln \frac{b^p(h)}{b^q(h)} \left(1 - \sum_{w \in S(h)} p(w|h')\right) + b^p(h) D(h') \\ &\quad - b^p(h) \sum_{w \in S(h)} p(w|h') \ln \frac{p(w|h')}{q(w|h')} \end{aligned} \tag{A.4}$$

Thus we are able to express $D(h)$, in terms of the n-grams explicitly defined in the LM.

We have used the tree-based representation of back-off n-gram models to derive the efficient computation scheme described above. An alternative approach for deriving the same relative entropy expressions presented above would be to consider n-gram back-off language models as a special case of Probabilistic Finite State Grammars (PFSG) (206) (207).

A.2 Incremental updates on a n-gram model

We now consider the calculation of incremental changes in R.E. between an in-domain n-gram back-off model p and an ML model q built from selected data. We are interested in finding out an efficient way to compute the change in R.E. when a sentence is added to the selected data set, thus changing the model q. Extending the notation from Section 2.2 let us define $C(wh)$ as the count of the word w seen with context h and $C(h)$ as the count for context h in the selected set (ML estimate $q(w|h) = C(wh)/C(h)$). We use $c(wh)$ and $c(h)$ to denote the counts in the current sentence. We assume that the model q has the same back-off structure as the model p. Thus we can divide $D(h)$ into just $T_1(h)$ and $T_4(h)$ depending on whether w is explicitly defined with context h in the model. We denote by $S(h)$, the set of words w for history h, for which the n-gram $p(w|h)$ is explicitly defined.

Constraining the update language model to have the same back-off structure as the in-domain model, we get from Equation (A.3)

$D(h) = T_1(h) + T_4(h)$

$$\begin{aligned} T_1(h) &= \sum_{w \in S(h)} p(w|h) \ln p(w|h) - \sum_{w \in S(h)} p(w|h) \ln q(w|h) \\ &= \sum_{w \in S(h)} p(w|h) \ln p(w|h) \\ &- \sum_{w \in S(h)} p(w|h) \ln \frac{C(wh)}{C(h)} \end{aligned}$$

After addition of a sentence the updated value of $T_1(h)$ is given by,

$$\begin{aligned} T_1^+(h) &= \sum_{w \in S(h)} p(w|h) \ln p(w|h) \\ &- \sum_{w \in S(h)} p(w|h) \ln \frac{C(wh) + c(wh)}{C(h) + c(h)} \\ &= T_1(h) + \ln \frac{C(h) + c(h)}{C(h)} \sum_{w \in S(h)} p(w|h) \\ &- \sum_{w \in S(h), c(wh) \neq 0} p(w|h) \ln \frac{C(wh) + c(wh)}{C(wh)} \end{aligned}$$

Thus the change in $T_1(h)$,

$$\begin{aligned} \delta T_1(h) &= \ln \frac{C(h) + c(h)}{C(h)} \sum_{w \in S(h)} p(w|h) \\ &- \sum_{w \in S(h), c(wh) \neq 0} p(w|h) \ln \frac{C(wh) + c(wh)}{C(wh)} \end{aligned}$$

The term $\sum_{w \in S(h)} p(w|h)$ can be precomputed since it is not a function of the word counts in the selected set.

We now consider $T_4(h)$ which we further split into two parts

$$\begin{aligned} T_4(h) &= \sum_{w \in S^c(h)} p(w|h) \ln p(w|h) \\ &- \sum_{w \in S^c(h)} b^p(h) p(w|h') \ln b^q(h) q(w|h') \\ &= \sum_{w \in S^c(h)} p(w|h) \ln p(w|h) \\ &- b^p(h) \underbrace{\sum_{w \in S^c(h)} p(w|h') \ln b^q(h)}_{T_{4A}(h)} \\ &- b^p(h) \underbrace{\sum_{w \in S^c(h)} p(w|h') \ln q(w|h')}_{T_{4B}(h)} \end{aligned}$$

Computing the change in $T_{4A}(h)$ ($\delta T_{4A}(h)$) requires computation of change in $b^q(h)$

$$\begin{aligned} T_{4A}(h) &= (1 - \sum_{w \in S(h)} p(w|h')) \ln b^q(h) \\ \delta T_{4A}(h) &= (1 - \sum_{w \in S(h)} p(w|h')) \delta \ln b^q(h) \end{aligned}$$

As for the $T_1(h)$ case, $\sum_{w \in S(h)} p(w|h')$ can be precomputed.

The expression for $b^q(h)$ is given by

$$\begin{aligned} b^q(h) = & \frac{1 - \sum_{w \in S(h)} \frac{C(wh)}{C(h)}}{1 - \sum_{w \in S(h)} \frac{C(wh')}{C(h')}} \\ & = \frac{C(h')}{C(h)} \frac{C(h) - \sum_{w \in S(h)} C(wh)}{C(h') - \sum_{w \in S(h)} C(wh')} \end{aligned}$$

The expression for $b^q(h)$ after the addition of a new sentence is given by,

$$\begin{aligned} b^{q+}(h) = & \frac{1 - \sum_{w \in S(h)} \frac{C(wh)+c(wh)}{C(h)+c(h)}}{1 - \sum_{w \in S(h)} \frac{C(wh')+c(wh^{'})}{C(h')+c(h')}} \\ & = \frac{C(h) + c(h) - \sum_{w \in S(h)} (C(wh) + c(wh))}{C(h') + c(h') - \sum_{w \in S(h)} (C(wh') + c(wh^{'}))} \\ & \times \frac{C(h') + c(h')}{C(h) + c(h)} \end{aligned}$$

Computation of change in $\ln b^q(h)$ ($\delta \ln b^q(h)$) is not required for the case where $c(h') = 0$. With $c(h') = 0$ we have $c(wh) = c(h) = c(wh^{'}) = c(wh) = 0$ and hence $b^{q+}(h) = b^q(h)$, which implies $\delta \ln b^q(h) = 0$.

For the case where $c(h') = 0$ and $c(h) \neq 0$, the computation of $\delta \ln b^q(h)$ is simplified. We have

$$\begin{aligned} \delta \ln b^q(h) &= \ln \frac{C(h') - \sum_{w \in S(h)} C(wh')}{C(h') + c(h') - \sum_{w \in S(h)} (C(wh') + c(wh^{'}))} \\ & \times \frac{C(h') + c(h')}{C(h')} \end{aligned}$$

$T_{4B}(h)$ can be expressed as,

$$\begin{aligned} T_{4B}(h) &= \sum_{w \in S^c(h)} p(w|h') \ln q(w|h') \\ &= -D(h') - \sum_{w \in S(h)} p(w|h') \ln q(w|h') \\ &+ \sum_{V} p(w|h) \ln p(w|h) \\ &= -D(h') - \sum_{w \in S(h)} p(w|h') \ln \frac{C(wh')}{C(h')} \\ &+ \sum_{V} p(w|h) \ln p(w|h) \end{aligned}$$

For $T_{4B}(h)$ the updated value after the addition of a new sentence can be expressed as,

$$T^{+}_{4B} = -D^{+}(wh') - \sum_{w \in S(h)} p(w|h') \ln \frac{C(wh') + c(wh^{'})}{C(h') + c(h')}$$

and thus the change in $T_{4B}(h)$, $\delta T_{4B}(h)$ can be expressed as

$$\begin{aligned} \delta T_{4B}(h) &= -\delta D(h') + \ln \frac{C(h') + c(h')}{C(h')} \sum_{w \in S(h)} p(w|h') \\ &- \sum_{w \in S(h), c(wh') \neq 0} p(w|h') \ln \frac{C(wh') + c(wh^{'})}{C(wh')} \end{aligned}$$

The total number of computations grows linearly with the total number of n-grams in the language model which grows exponentially with the order of the model. For initialization, we use a unigram model initialized with a random subset of data to seed data selection (Section 2.2). The data selected with the unigram model is then used to initialize the counts for the q model.

Appendix B

PI meeting presentation

USC SpeechLinks R&D Updates

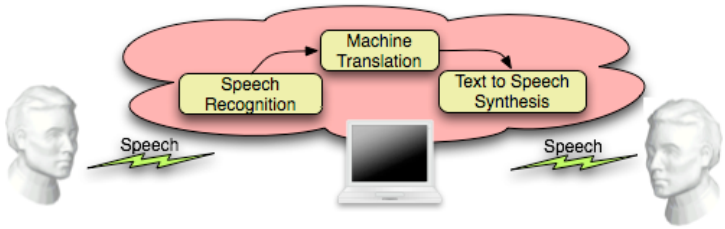
Shri Narayanan
Panos Georgiou

USC Viterbi School of Engineering
Speech Analysis and Interpretation Lab
November 14, 2007

S2S Current Technologies and Future Emphasis

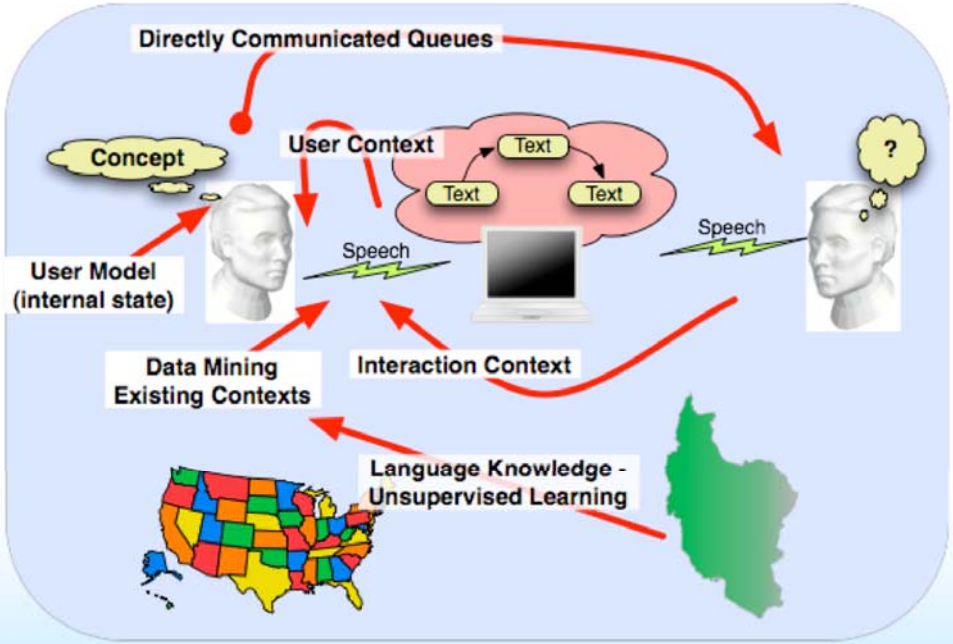
Current work: most emphasis on Component Technologies

- Good Progress, but nearing the bounds of what is possible
- Communication is context specific
- Lack of “interpretation”
- Language resources are hand-crafted — expensive and tedious
- Good work “inside the boxes” but no integrated component optimization



Human Communication is Complex

- Need for:
  - Human centric design
  - User and Interaction Context
  - Rapid language knowledge exploitation
  - Next-gen bio-inspired models of speech production/perception
  - Tighter component integration



USC SpeechLinks: Highlights

Our Goal

- ***Human Aware Interpretation***
  - *System aware of interaction*
  - *System aware of both the users*
- ***Rapid Development***
  - *Targeted, scenario driven collections*
  - *Self learning systems*
- ***Tighter Integration***
  - *Tight integration of components to create a true system, rather than just an assembly of component technologies*

Highlights

- **Data collection:**
  - 50+ hours of 2 way Farsi-Farsi communication
  - Internal formalization efforts
  - Several S2S collections (most recent 5-setup, 50 interactions)
- **New modular system**
  - Everything (except TTS) redone
  - Runs on Windows, Linux, OS X, (iPhone?)
- **Data mining**
  - 24% relative improvement on LVCSR tasks
- **Linguistic Knowledge**
  - **Colloquial 2 Formal mapping 12% BLEU improvement**
- **Topic models**
  - Over 20% perplexity reduction, 2% absolute WER reduction
- **User aware processing**
  - Improved user satisfaction
  - Improved user behavior
  - Multimodal benefits
- **Bio-Inspired Speech models**
  - Robustness to noise
  - Our proposed methods outperforms MFCC and RASTA by 20-40%
- **Over 25 publications in the last year**

Behind the Scenes

- Work on several fronts, many of which have not been shown in the live system:
  - Front end:
    - Bio-inspired features
    - Prosody in language
  - Language modeling:
    - Data mining
    - Clustered Parallel LM's
  - Machine Translation:
    - Normalization
    - Morphology
  - User modeling:
    - User type
    - Multi- vs uni-modal
    - User learning curve
  - System description

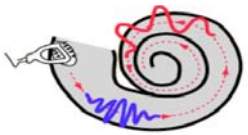
Early Auditory Processing Inspired Features for Robust Speech Recognition (InterSpeech 2007, EUSIPCO2007)

Motivation:

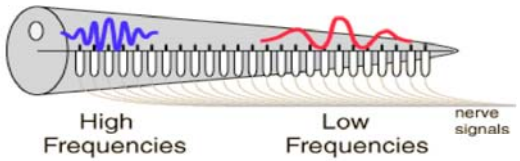
- The human auditory system (HAS) can robustly *select* (localize), *segment*, and *recognize* sounds/speech embedded in complex scenes.
- Machine performance degrades drastically in various conditions: noise, speaker changes, overlapping sources.

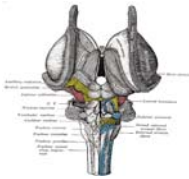
Goal:

- To reduce the performance gap between human and machines by understanding and modeling the information processing stages in HAS



Cochlea





Brain Cochlear Nucleus

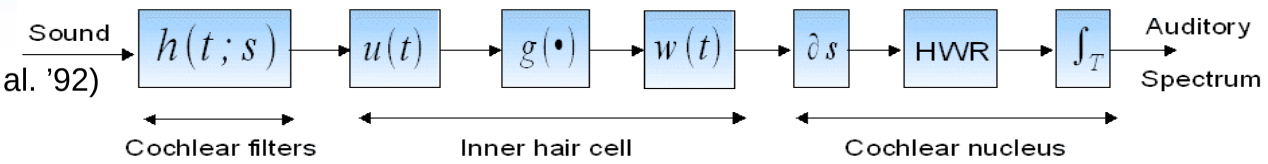
Contribution and findings:

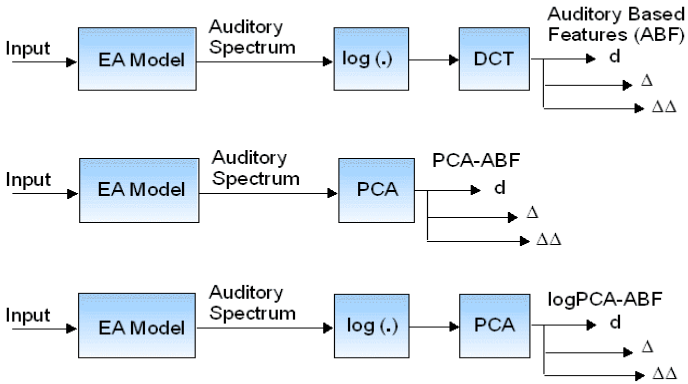
- **Derived bio-inspired features for ASR based on the processing stages in the early human auditory system**

- **New set of features provided substantial improvement over both state of the art MFCC (mel frequency cepstral coefficients) and RASTA (RelAtive SpecTrA) features in noisy conditions**

Early Auditory Processing Inspired Features for Robust Speech Recognition (InterSpeech 2007, EUSIPCO2007)

The EAM (Yang et al. '92)



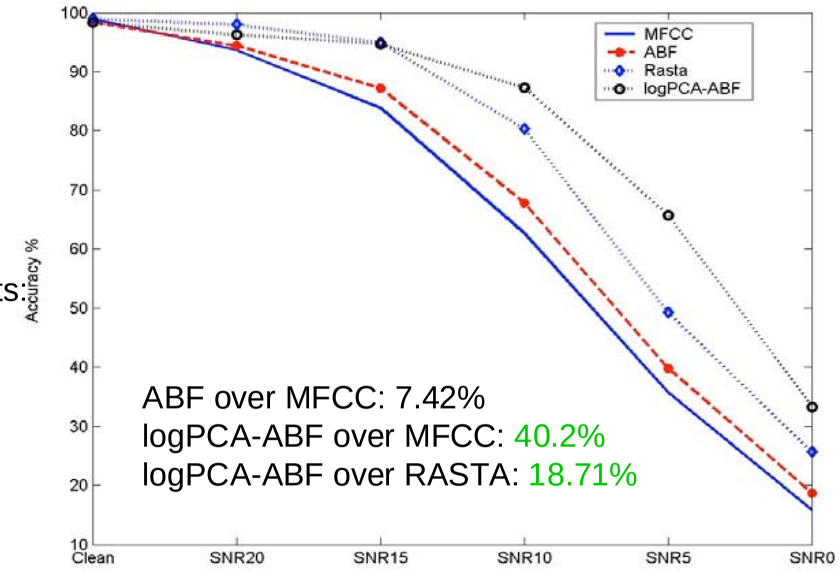


(a) **ABF**: Replace Mel-filterbank with a more accurate EA model

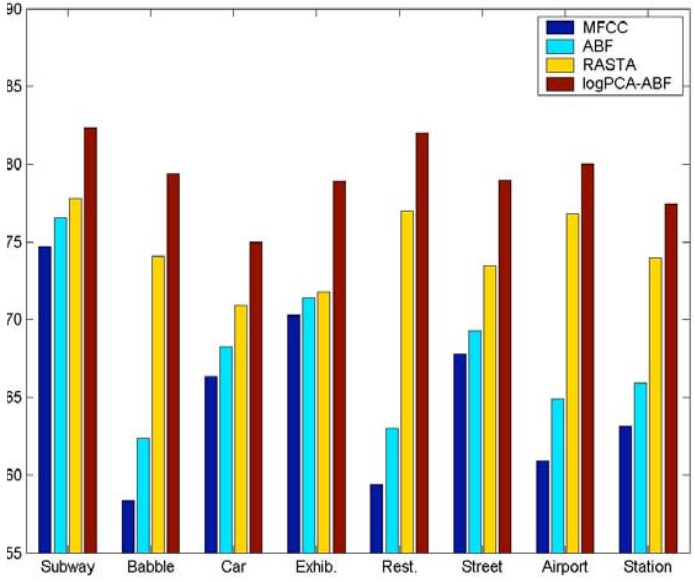
(b) **PCA-ABF**: Retain only the cochlear channel outputs that are more likely to fire neurons in CAS

(c) **logPCA-ABF**: Modify EA model to incorporate the non-linear compression due to the OHC and then pursue the idea in (b)

Aurora Results:
MFCC,
ABF,
RASTA,
logPCA-ABF



* The average of word accuracy over all noise types for each SNR level



* The average of word accuracy over all SNR levels for each noise type

Direct Articulatory Measurements for Speech Recognition

(ICASSP2007)

Motivation:

- Articulatory data capture fine details of the underlying speech production
- **Signal Representation**: Evidence from the articulatory stream may be useful to strengthen our ability to discriminate speech sounds

Open Questions:

- To what extent articulatory features provide complementary information for recognizing speech sounds ("phones")?
- What is the optimal way to perform feature extraction for the articulatory signal?

Contribution and Findings:

- **Demonstrated the complementary nature of direct articulatory evidence with respect to acoustic data through Mutual Information calculations**
- **Proposed novel time and frequency based signal representations (Parsimonious discriminative Articulatory Features)**
- **Derived optimal analysis window length to maximize phone discrimination offered by the articulatory data, again using Mutual Information**

Prosody in spoken language processing

(several publications)

Motivation:

- To improve translation move from “word” recognition to include richer representations:
  - Suprasegmental processing: representation, detection & modeling of prosody

Goals:

- Capture and transfer information in spoken language beyond just words
  - Currently translation systems rely on just textual representations
  - Goal is to transfer meta-information such as prosody, segmentation within the s2s framework.
- Automatic detection of prosody (pitch accent, boundary tones, rate, prominence)

Contributions:

- Q/A detector *(real time implementation, not integrated in SpeechLinks yet)*
- New algorithms/features for supervised & unsupervised automatic prosody labeling: Maximum entropy models, Coupled HMM models Fuzzy and GMM clustering
- State-of-the-art results in prosody detection
- Novel bio-inspired Saliency based auditory model to detect prominence in speech
- Dialog act tagging using Maximum entropy model: Uses n-grams of prosodic observation sequence, state of the results
- Improving Automatic speech recognition: rescoring lattices with prosodic information

Data issues

- Many variants of same words like ‘did you’ ‘didja’ in English
- Informal words sometimes joined, just one case “rA” -> “_v” potentially doubles noun vocabulary.
- What we tried:
  - Colloquial to formal conversion:
    - Significant reduction in vocabulary
    - Improvement in BLEU
    - Issues with vocabulary based approach on one-many mappings
    - Combination of rules and table-lookup to normalize training data and input.
    - **MT: 12% improvement en->fa, 6% fa->en**
  - Removal of homographs:
    - 1-1 mappings: dictionary based
    - 1-many mappings:
      - exploit context
      - exploit meaning in parallel text (some success)
    - Many-many: need some manual intervention and we run out of resources!
    - In the end we followed a frequency based approach at the tts output (many concepts = one pronunciation as Appen data, but with frequency considerations)
      - This can be really problematic in a some cases. It can make a small number of certain concepts deterministically impossible to translate. Example: negation vs no

Data issues

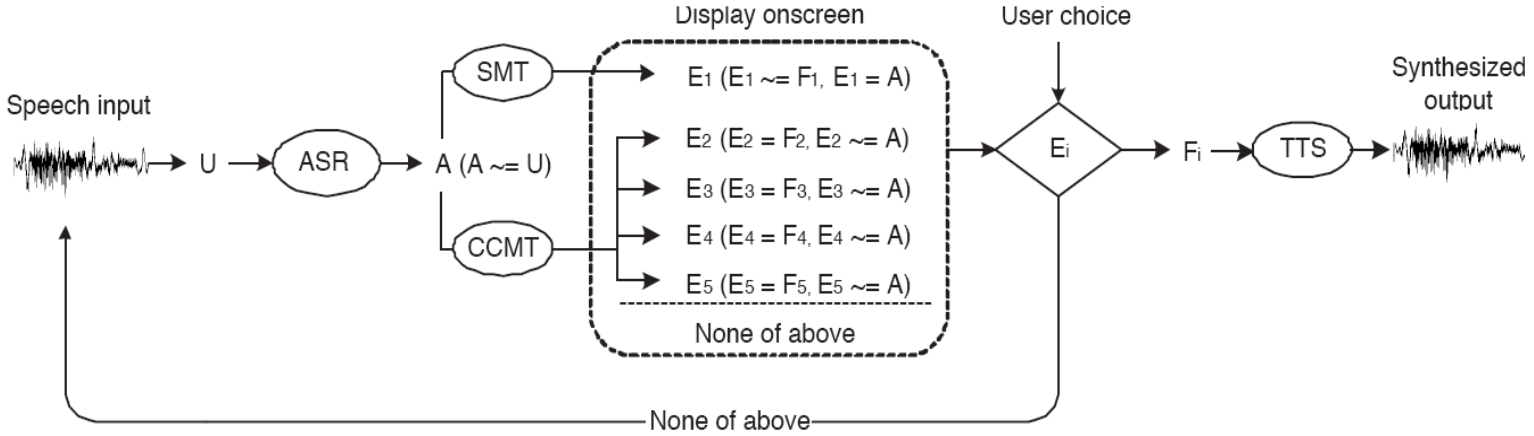
- Problems with Persian's complex word structure
  1. Increases vocabulary, e.g. a Persian verb generally has several **dozen** forms.
  2. One Persian word may correspond to several English non-contiguous words (breaks phrase MT)
  3. Highly ambiguous En->Fa (Fa->En 54% better!)
- Work in progress:
  - Segment words to lower vocab (modest boost)
  - Morphological analysis

Data mining and topic LM's

- **Motivation:**
  - Machine performance drops significantly out of domain
  - Domain definition vague and non-uniformly covered
  - Just making LMs larger damages regions of good coverage (such as medical domain)
- **Goals**: better language modeling
  - Better language models
  - Better coverage of uneven topics
- **Current work highlights:**
  - [building on past work] Mining:
    - 15-20% relative performance improvement in word error rate (WER) on limited domain systems (Transonics, Boston radio)
    - Best system in the 2007 TC-STAR English ASR evaluation (with IBM)
      - Achieved word error rate of 7.4% compared to 9.6% of IBM system
      - Second best submission had WER of 9.5%
      - Combining output from all competing sites gave 7.2%WER
  - Cluster LM creation: new results (PPL reductions relative to DARPA training data):
    - Training+web: **8% reduction**. Training + web + clustering LM's: **24% reduction** (STILL 1 final LM)
  - Topic LM rescoring with just 2 LM's gives ~2% absolute improvement

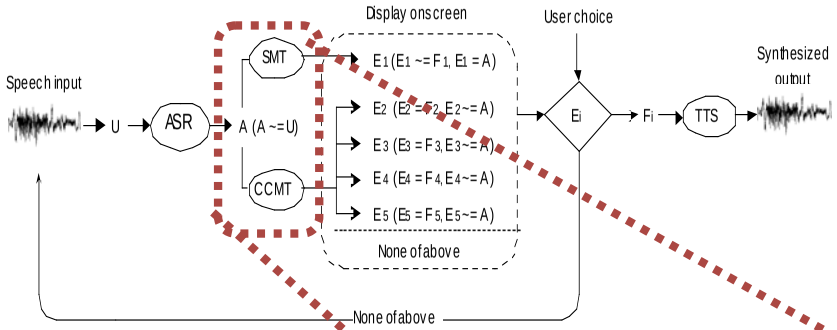
User-centric Design for Robust Translingual Communication

- **Motivation:**
  - we observe a large variance in s2s communication success, with
  - constant average machine performance
  - Many of the errors are caused by users (“and” do you ....)
- **Goals:** Devices that can work *with* humans adaptively
  - Design and evaluate interface options including the role of multimodality in concept transfer.
  - Design intelligent, user-adaptive machine mediation to enhance concept transfer. Exploit user behavior modeling and prediction.
  - Design, and evaluate multiple mechanisms of S2S interactions to optimize the best user awareness and context awareness settings for a given domain condition
- **Current work highlights:**
  - Multimodal interface versus a Unimodal interface work. Best paper award MMSP 2007
  - User agent: Modification of user patterns towards more acceptable system operating regions. 10% increase of operation within the normal usage region with feedback.
  - Benefits of user learning, through longitudinal studies.



Internal procedure of SpeechLinks (Choice mode)

ASSUME YOU ARE A ENGLISH SPEAKER THAT DOESN'T KNOW FARSI



Display: I can try and translate:
thank you very much
Will translate: *I thank you many much*

Very going to much is an invisible error

thank you very much

SMT

A (A ~= U)

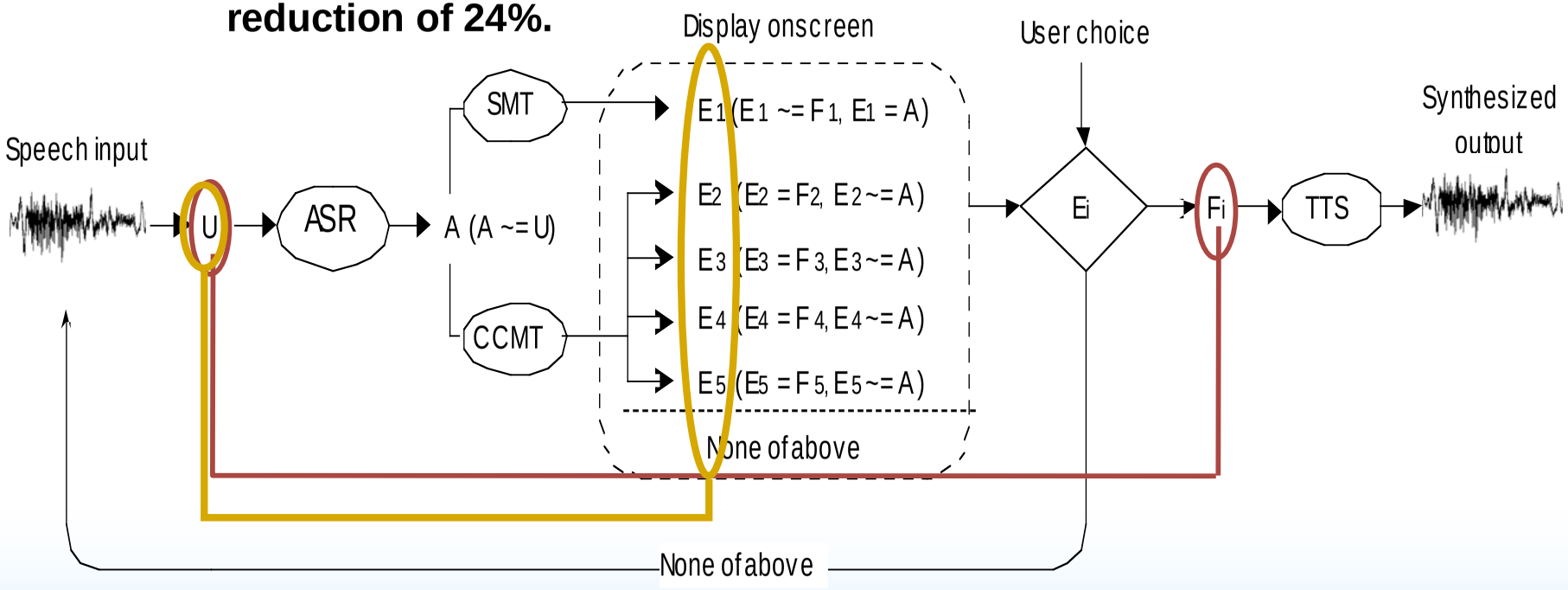
CCMT

All errors and "quantizations" are visible to user

Display: **Thank you**
Will Translate: *Thank you*

Multimodal behavior evaluation

- Hypothesis 1:
  A multimodal interface, employing both the audio and text modalities, will be better than a single-modality interface utilizing audio only, in terms of translation quality.
- Results
  - User choice through the use of the visual modality gives a **relative error reduction of 24%.**



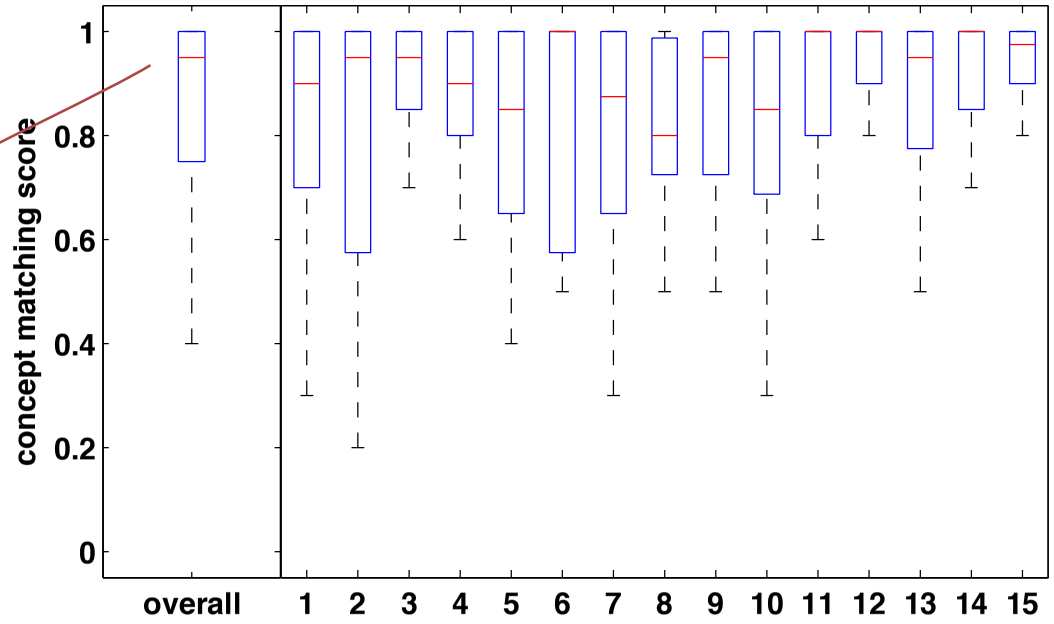
Multimodal behavior evaluation

- Hypothesis 2:

  Users will accept certain errors from the utterances provided by the system. The degree of degradation in terms of concept representation that different users are willing to accept varies.

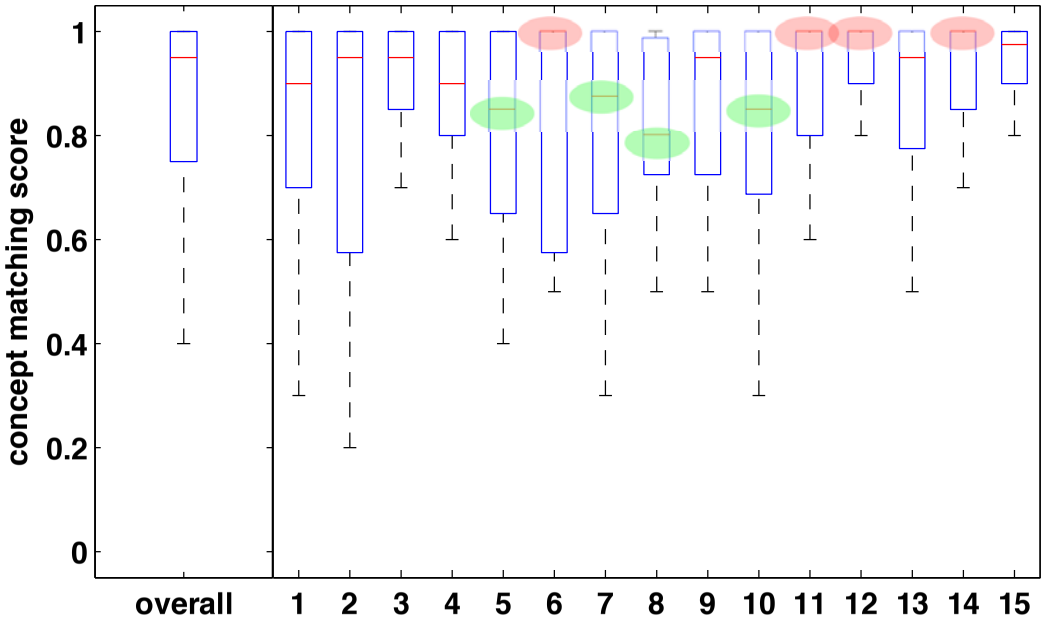
- Results
  - The **median score of 0.95** indicates that half of the time the users accepted the onscreen machine-produced utterances when they contained 95% of the concepts in the original utterances (std. dev. 0.21)
  - Users accept machine-produced utterances with CMS as low as 0.4, which reveals quite accommodating behavior.
  - The mean concept matching score was 0.84, indicating that users on average are accepting of 16% concept loss.

Results: Hypothesis 2



- The **median score of 0.95** indicates that half of the time the users accepted the onscreen machine-produced utterances when they contained 95% of the concepts in the original utterances (std. dev. 0.21)
- Users accept machine-produced utterances with CMS as low as 0.4, which reveals quite accommodating behavior.
- The mean concept matching score was 0.84, indicating that users on average are accepting of 16% concept loss.

Results
(2) User Error Tolerance



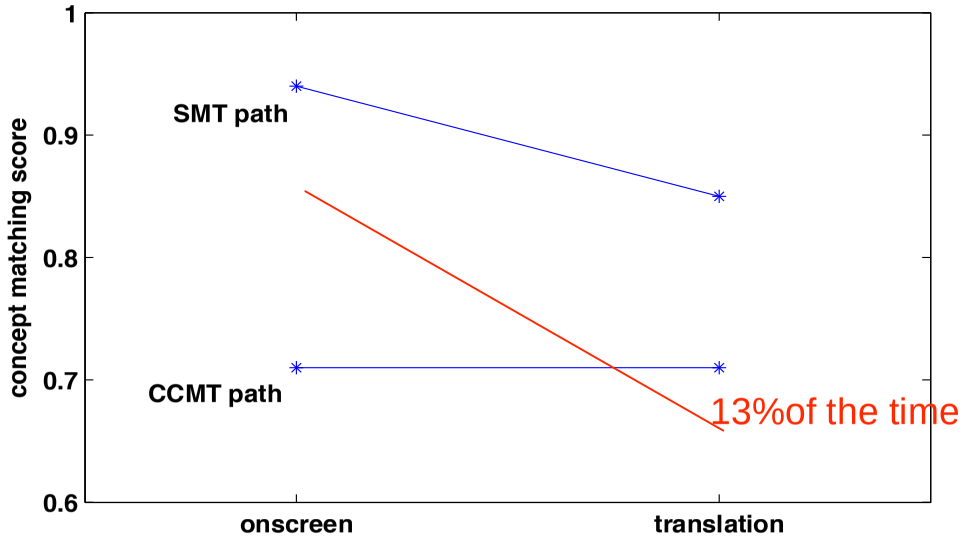
- Users in the interactions, 6, 11, 12, 14 were picky in accepting machine produced utterances;

- Users in interactions, 5, 7, 8, 10 were more accommodating than others in acceptance of concept errors in the utterances produced by the system.

- Users in interactions 3,12,13,15 were more consistent in accepting concept errors than users in interactions 1,2,7,10.

Multimodal behavior evaluation

- Hypothesis 3: **When incorporating multiple types of translation method for developing a push-to-talk interface, appropriate feedback is required to guide the users in their choices of translation method.**
- Results
  - (Note that the concept classifier has no subsequent loss after the user choice - it's 'backtranslation' is very accurate)
  - We found that in 13% of the data the users decided to take through the SMT path, performance would have been improved if the users had chosen the CCMT path → Discrepant translation quality

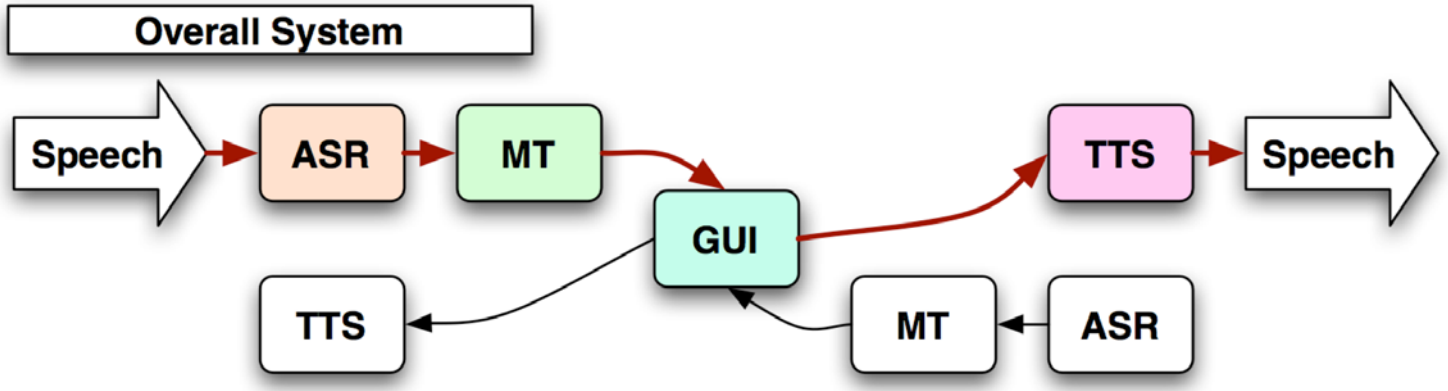


Longitudinal studies

- Hypothesis:
  - Users learn more appropriate translation strategies over time
- Experiment description:
  - 4 teams of English and Farsi speakers using the S2S system
  - 8 sessions over 4 weeks per team
  - 12 medical domain scenarios
  - User interview, survey questionnaires, video, and log data
  - Used human tagging (concept matching scores)
- Results:
  - Increasing user-perceived learning curve during the 4 weeks of experiment.
  - Improved user strategies in accommodating to errors
  - Improved strategies in preventing errors

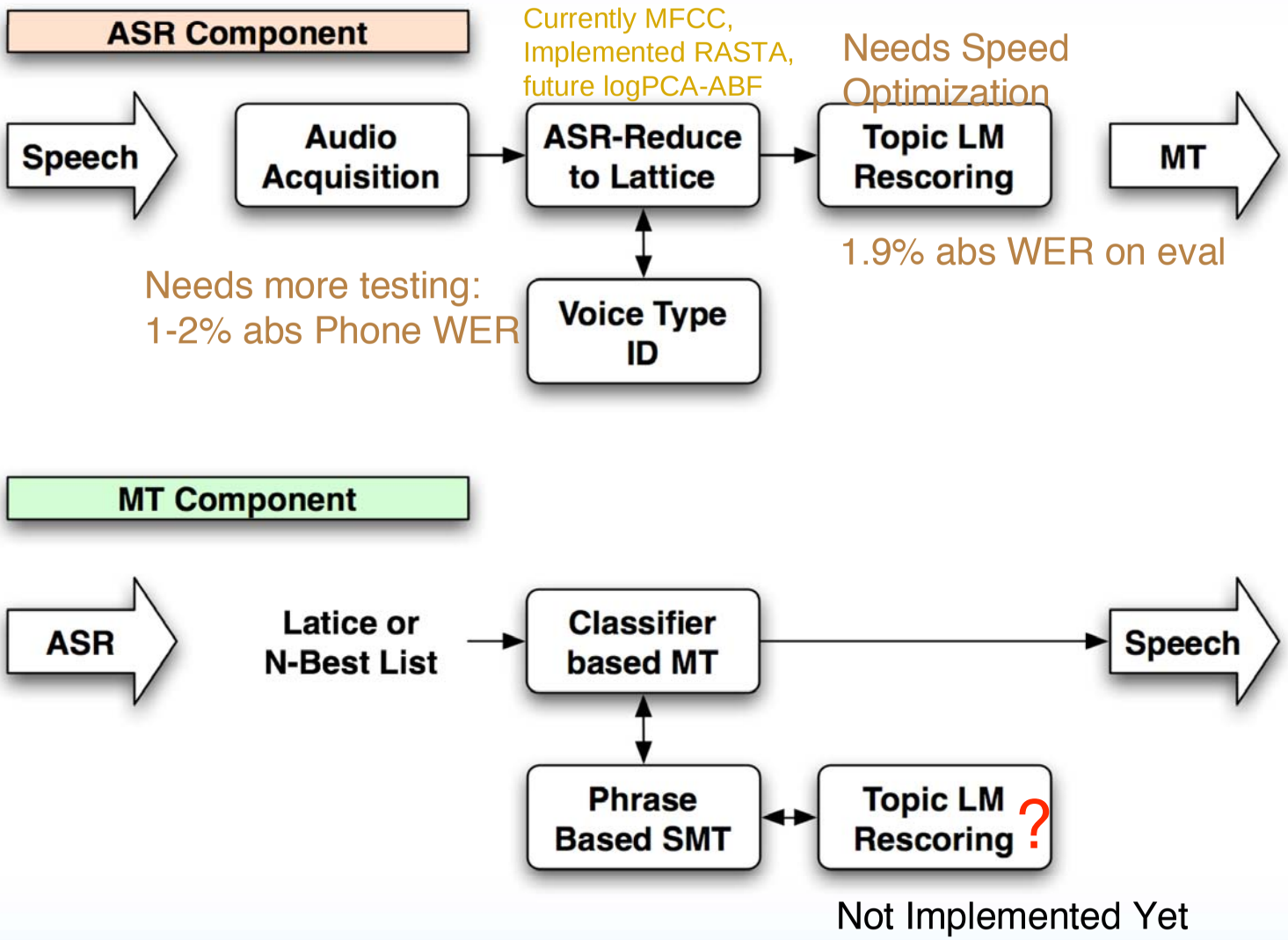
System Developments

- Over last 12 months completely re-programmed the whole system:

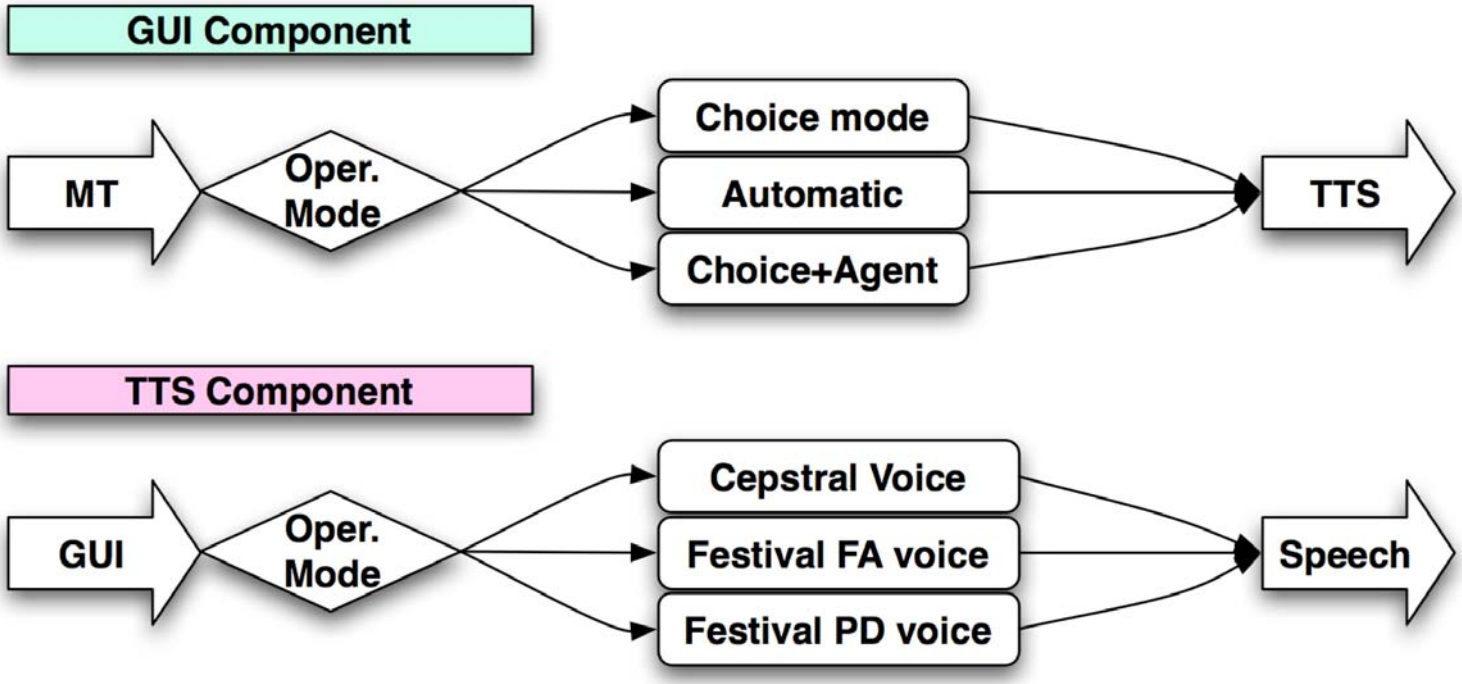


- Communication platform (slserver)
- ASR (new engine OtoSentia)
- SMT (new engine OTP)
- TTS: still using Festival &/or Cepstral

Components


ASR Component
Currently MFCC, Implemented RASTA, future logPCA-ABF
Needs Speed Optimization
Speech
Audio Acquisition
ASR-Reduce to Lattice
Topic LM Rescoring
MT
1.9% abs WER on eval
Needs more testing:
1-2% abs Phone WER
Voice Type ID
MT Component
ASR
Latice or N-Best List
Classifier based MT
Speech
Phrase Based SMT
Topic LM Rescoring ?
Not Implemented Yet

Components$_2$



- Choice mode & agent make more sense in collaborative task based interactions

USC SpeechLinks: Highlights

Our Goal

- ***Human Aware Interpretation***
  - *System aware of interaction*
  - *System aware of both the users*
- ***Rapid Development***
  - *Targeted, scenario driven collections*
  - *Self learning systems*
- ***Tighter Integration***
  - *Tight integration of components to create a true system, rather than just an assembly of component technologies*

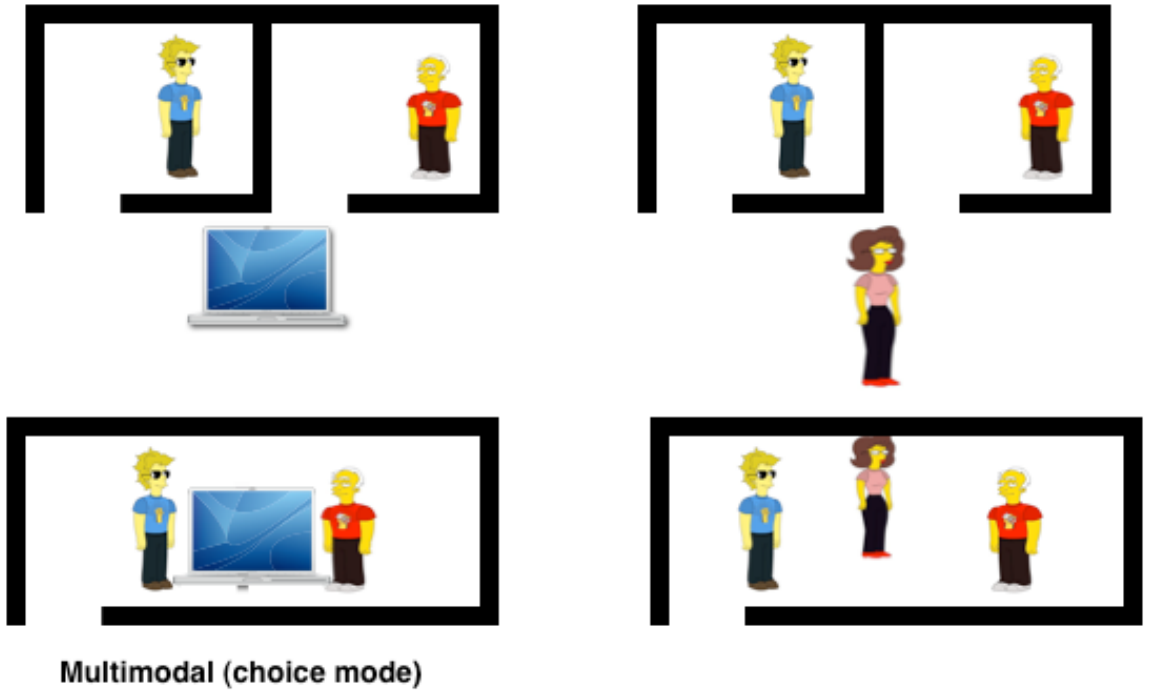
Highlights

- **Data collection:**
  - 50+ hours of 2 way Farsi-Farsi communication
  - Internal formalization efforts
  - Several S2S collections (*most recent 5-setup, 50 interactions*)
- **New modular system**
  - Everything (except TTS) redone
  - Runs on Windows, Linux, OS X, (iPhone?)
- **Data mining**
  - 24% relative improvement on LVCSR tasks
- **Linguistic Knowledge**
  - Colloquial 2 Formal mapping 12% BLEU improvement
- **Topic models**
  - Over 20% perplexity reduction, 2% absolute WER reduction
- **User aware processing**
  - Improved user satisfaction
  - Improved user behavior
  - Multimodal benefits
- **Bio-Inspired Speech models**
  - Robustness to noise
  - Our proposed methods outperform MFCC and RASTA by 20-40%
- **Over 25 publications in the last year**

THANK YOU!

Current experiments: 5-way modality comparison

- Underway:
  - 50 sessions of



Multimodal (choice mode)

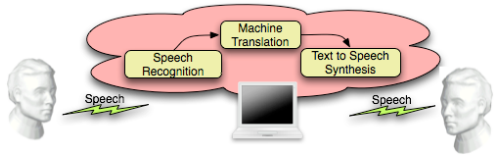
Unimodal (machine decides)

USC SpeechLinks: Tactical Cross-Lingual Communication

STATUS QUO

Emphasis on Component Technologies

Good Progress, but nearing the bounds of what is possible



- *Communication is context specific*
- *Lack of "interpretation"*
- *Language resources are hand-crafted — expensive and tedious*
- *Good work "inside the boxes" but no integrated component optimization*

NEW INSIGHTS

Human Communication is Complex

Need for: **Human centric design, User and Interaction Context, Rapid language knowledge exploitation**, next-gen **bio-Inspired** models of speech production, tighter component integration

USC-SPEECHLINKS ACHIEVEMENTS

MAIN ACHIEVEMENTS:

- **Novel modular, distributable system architecture**
- **User models**: of interpretation strategies for mis-translations/ communication uncertainty (**Best paper award**, IEEE Multimedia Signal Processing 2007)
- **Automated resource exploitation**: language models through data mining (Key Contributions to Best system performance in TC-STAR 2007)
- **Exploiting unsupervised linguistic knowledge.**
- **Context sensitive processing:** ability to handle interaction within statistical dialog context
- **Target, Scenario Driven Data Collections** for Surprise Language

HOW IT WORKS:

- User models identify the level of accommodation and expertise of the user, and provide real-time training on achieving optimal cross-lingual communication
- Automated exploitation of existing language resources by identifying data with high distributional similarity to the target language the system needs. System then can **self-train** and improve over usage through daily self-updates
- Statistically representing the context transition probabilities among language-model representations of dialog states, allows for disambiguating context

ASSUMPTIONS AND LIMITATIONS:

- Many of our research developments are proven and used outside the TRANSTAC realtime system due to the lack of resources to implement robust real-time software for these. They are robust and proven in faster machines or non-real time settings and can be transitioned into the future realtime system version.

QUANTITATIVE IMPACT

Data mining provides 5% improvement on challenging domain sets.

Linguistic Knowledge gains 37.5% supervised/11.5% semi-supervised +6.5% unsupervised

Context models: Employ data mining techniques for statistical interaction context models.

User Aware processing provides 24% concept transfer improvement.

Bio-Inspired Speech models: Enable more robust speech recognition in noisy environments (18-40% improvement at feature level)

END-OF-PHASE GOAL

Human Aware Interpretation

- *System aware of interaction*
- *System aware of both the users*

Rapid Development

- *Targeted, scenario driven collections*
- *Self learning systems*

Tighter Integration

- *Tight integration of components to create a true system, rather than just an assembly of component technologies*

Bridging the Communication Gap is vital for Global Tactical Operations